VŠB – Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

Katedra telekomunikační techniky

Metodika návrhu číslicových filtrů IIR a FIR

Metodology of IIR and FIR Digital Filters Design

2012 **Bc. Vratislav Pech**

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání diplomové práce

Student: **Bc. Vratislav Pech**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2601T013 Telekomunikační technika

Téma: Metodika návrhu číslicových filtrů IIR a FIR
Metodology of IIR and FIR Digital Filters Design

Zásady pro vypracování:

1. Základní vlastnosti číslicových systémů. Principy lineární filtrace.
2. Popis metod návrhu IIR filtrů, zejména pomocí Cauerovy aproximace.
3. Popis metod návrhu FIR filtrů, zejména optimalizační metody.
4. Analýza vlivu konečné délky slova na navržený číslicový filtr.
5. Vytvoření grafického uživatelského rozhraní pro návrh IIR a FIR filtrů v MATLABu s možností vizualizace jednotlivých kroků návrhu.

Seznam doporučené odborné literatury:

SMÉKAL, Z. *Číslicové filtry*. Vydání 1. Brno: Nakladatelství VUT, 1993. 136 s. ISBN 80-214-0500-7.

DAVÍDEK, V.; LAIPERT, M.; VLČEK, M. *Analogové a číslicové filtry*. Vydání 2. Praha: Nakladatelství ČVUT, 2006. 345 s. ISBN 80-01-03026-1.

ZAPLATÍLEK, K.; DOŇAR, B. *MATLAB: tvorba uživatelských aplikací*. Vydání 1. Praha: BEN - technická literatura, 2004. 215 s. ISBN 80-7300-133-0.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Jan Skapa, Ph.D.**

Datum zadání: 19.11.2010
Datum odevzdání: 04.05.2012

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA
Fakulta elektrotechniky a informatiky
OSTRAVA

prof. RNDr. Vladimír Vašinek, CSc.
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem diplomovou práci na téma Metodika návrhu číslicových filtrů IIR a FIR vypracoval samostatně s použitím odborné literatury a pramenů, uvedených na seznamu, který tvoří přílohu této práce.

V Praze 15.8.2012

....

(podpis studenta)

Poděkování

Tímto bych chtěl poděkovat svému vedoucímu práce, Ing. Janu Skapovi, Ph.D., za cenné rady, připomínky a náměty. Rovněž mu děkuji za trpělivost, kterou mi během konzultací věnoval a to především v období mimo výuku.

Abstrakt a klíčová slova

Abstrakt

Tato práce se zabývá metodikou návrhu číslicových filtrů IIR a FIR. Cílem práce bylo stručně popsat současný stav problematiky návrhu filtrů a následně vytvořit GUI pro návrh IIR a FIR filtrů v prostředí MATLAB. Byl zde kladen důraz na vizualizaci jednotlivých kroků návrhu a uživatelskou přívětivost. Uživatel tak může pomocí myši a klávesnice nastavovat vstupní parametry a ovládat jednotlivé kroky návrhu a také snadno ukládat výsledné hodnoty.

Klíčová slova

GUI, MATLAB, IIR, FIR, Cauer, REMEZ, návrh filtru, kvantování, aproximace

Abstract

The thesis contains metodology of IIR and FIR digital filters design. The aim of the thesis was to briefly describe the current state of design issues associated with these filters and then to create the graphical user interface for digital filter design using MATLAB. There was an emphasis on visualization of individual steps of the design and on user friendly interface. By using mouse and keyboard user can set input parameters and control individual steps of the design and so easily safe output values.

Key words

GUI, MATLAB, IIR, FIR, Cauer, REMEZ, filter design, quantization, approximation

Obsah

Seznam použitých symbolů a zkratek

IIR	Infinite impulse response - nekonečná impulsní odezva
FIR	Finite impulse response - konečná impulsní odezva
LTI	Linear time-invariant - lineárně časově invariantní
MATLAB	Matrix Laboratory - programové prostředí
GUI	Graphical user interface - grafické uživatelské rozhraní
GUIDE	GUI development environment - sada nástrojů pro vytváření GUI
NDP	Normovaná dolní propust
Hertz	Jednotka frekvence

1 ÚVOD

Diplomová práce se zabývá metodikou návrhu číslicových filtrů. Cílem bylo se seznámit se základními vlastnostmi číslicových systémů, metodikou návrhu filtrů a vlivy konečné délky slova.

V první části je stručně popsána problematika a vlastnosti číslicových systémů. V další části jsou rozebrány číslicové filtry, Cauerova aproximace a stejnoměrně zvlněná optimalizační metoda. Důležité je zde vzájemné srovnání filtrů IIR a FIR a výčet jejich základních vlastností. V kapitole týkající se kvantování koeficientů je mimo jiné provedeno stručné porovnání prezentace čísel v přesnosti Single vs. Double. Poslední část pojednává o tvorbě aplikací IIR a FIR a popisuje uživatelské možnosti. Je zde kladen důraz na jednoduchost a transparentnost návrhu, aby i méně znalí uživatelé mohli ihned začít s návrhem.

2 Základní charakteristiky číslicových signálů.

Číslicový signál je signál diskrétní, tzn. jeho hodnoty leží v určité konečné množině hodnot. Velikost množiny úzce souvisí s počtem možných hladin vzorků, které musí ležet v této množině. Hodnoty ležící mimo tuto množinu jsou zaokrouhleny k nejbližší možné. Proces kvantování způsobí ztrátu informace a dojde tedy k určitému nevratnému zkreslení a do signálu je natrvalo vnesen tzv. kvantovací šum.

Číslicové signály mají následující dvě vlastnosti:

1. jsou to signály s diskrétním časem
2. jejich množina možných hodnot je konečná, tzn. je diskrétní v hodnotách.

Tato diskretizace je jak v čase, tak v hodnotách.

Signály můžeme reprezentovat posloupnostmi čísel mající konečný počet prvků.

V technické praxi používáme číslice 0 a 1. Nejen číslicové systémy, ale i číslicové signály mají své výhodné vlastnosti obzvláště z hlediska přenosu zpráv. Jednotlivé číslice jsou dobře rozeznatelné a to i po jejich značném zkreslení i po přidání aditivního šumu. Číslicově vyjádřené zprávy lze tak snadno utajovat pomocí šifrování nebo zabezpečit proti chybám při přenosu, je možné potlačovat nadbytečnou informaci ve zprávách. Při použití číslicového vyjádření s dostatečným počtem vzorků lze dosáhnout rozsáhlé dynamiky hudebního signálu.

2.1 Základní operace s číslicovými signály

Číslicové systémy používají tři základní lineární operace, které můžeme realizovat použitím sčítačky, násobičky a zpožďovacího členu. Tyto tři prvky jsou základním stavebním kamenem pro realizaci lineárních číslicových filtrů (IIR, FIR viz kap. 3 a 4).

Součin signálů

Signály přivedené na vstup vynásobí libovolnou reálnou konstantu.

Součet signálů

Signály přivedené na vstup jsou jednoduše sečteny.

Zpoždění signálů

Realizujeme jej pomocí posuvných registrů nebo využitím příkazu cyklu v programu. Zpravidla využíváme posun posloupnosti vzorků o jeden vzorkovací krok.

2.2 Základní vlastnosti číslicových systémů

Mezi základní vlastnosti číslicových systémů řadíme linearitu, stabilitu, kauzalitu a časovou invariantnost soustavy.

Linearita

Platí zde princip superpozice. Číslicový systém je lineární, jestliže pro lineární kombinaci vstupních signálů je výstupní signál tvořen lineární kombinací odezev na dílčí vstupní signály.

Stabilita

Pokud přivedeme na stabilní systém jakýkoliv omezený signál, bude reagovat omezeným výstupním signálem, pokud však je systém stabilní. Neboli pokud je odezva na omezený vstupní signál opět omezená. Dále platí, že součet absolutních hodnot vzorků impulzní charakteristiky bude také konečný.

Kauzalita

Pokud je systém kauzální tak závisí pouze na předešlých vstupních vzorcích a nezávisí na vzorcích budoucích. Neboli kauzální systém může pracovat pouze v reálném čase.

Časová invariantnost

Pokud přivedeme na vstup signál s určitým časovým posunem tak na výstupu dostaneme signál se stejným časovým posunem. Systém zachovává vlastnosti a hodnoty svých parametrů v čase.

2.3 Popis LTI spojitých systémů

V těchto systémech platí základní vlastnosti obdobně jako v již zmíněných číslicových systémech. Lze je popsat diferenciální rovnicí nebo impulzní odezvou.

2.4 Popis LTI diskrétních systémů

Tyto systémy lze popsat dvěma způsoby:

- v časové oblasti: diferenční rovnice a impulzní odezva
- ve frekvenční oblasti: přenosová funkce, frekvenční charakteristika (modulová a fázová)

2.5 Lineární filtrace

Lineární filtrace patří do lineárního časově invariantního systému, takže splňuje princip superpozice. Filtraci lze chápat jako zpracování signálu sloužící k výběru určitých složek z více signálů, které ponecháme nebo zesílíme, a k potlačení ostatních složek, často nežádoucích. Nejčastěji se setkáváme s filtrací ve frekvenční oblasti, kde tyto harmonické komponenty – amplitudy a fáze se pozmění. U diskrétních signálů jsou charakteristiky periodické, takže stačí udávat jejich hodnoty v rozsahu kmitočtů $\langle 0, F_{vz}/2 \rangle$, kde F_{vz} je vzorkovací frekvence. Uvedené vlastnosti využívají lineární filtry, které můžeme rozdělit na základě délky impulzní odezvy na filtry FIR a IIR.

V kapitole 2 bylo čerpáno z: [2],[3]

3 Popis metod návrhu IIR filtru

3.1 Úvod do IIR filtrů

Číslicové filtry s nekonečnou impulsní charakteristikou (infinite impulse response) zkráceně IIR, využívají pro realizaci zpětných vazeb (rekurzí), tj. rekurzivních struktur.

Dle fáze je lze rozdělit na filtry:

- s minimální fází (všechny nuly přenosu leží uvnitř jednotkové kružnice)
- s maximální fází (všechny nuly přenosu jsou vně jednotkové kružnice)
- se smíšenou fází (některé nuly se nachází vně, jiné uvnitř)

Diferenční rekurzivní rovnice mezi vstupem x_n a výstupem y_n je ve tvaru:

$$y_n = \sum_{i=0}^{r} L_i x_{n-i} - \sum_{i=1}^{m} K_i y_{n-i} \tag{3.1}$$

L_i, K_i jsou systémové koeficienty v dopředných či zpětných vazbách, r udává počet zpoždění v nerekurzivní složce rovnice, m je řád systému (obvykle platí $m \geq r$) a rovněž počet zpoždění pro rekurzivní složku.

Přenosová funkce (obrazový přenos) je v součtovém či součinovém tvaru racionální lomená funkce, a je dána z-transformací impulsní charakteristiky filtru:

$$H(z) = \frac{\sum_{i=0}^{r} L_i z^{m-i}}{\sum_{i=0}^{m} K_i z^{m-i}} = A \frac{\prod_{i=1}^{r} (z - n_i)}{\prod_{i=1}^{m} (z - p_i)}, \text{ platí } K_0=1. \tag{3.2}$$

Podíl těchto polynomů v proměnné z tvořených konstantami L_i a K_i lze rozložit na součiny kořenových činitelů. Tímto získáme pro charakteristiku systému konfiguraci čítající: zesílení A, póly p_i a nulové body n_i. Póly realizují rekurzivní část, nulové body nerekurzivní část.

Frekvenční charakteristika filtru je komplexní a periodická, a může být vyjádřena z rovnice obrazového přenosu po dosazení za $z=e^{j\omega T}$ a má tento výsledný tvar:

$$G(\omega) = H(e^{j\omega T}) \tag{3.3}$$

3.1.1 Základní realizace IIR filtrů

Vhodná volba struktury filtru obvykle rozhoduje o výpočetní náročnosti a dalších jevech spojených s aritmetickou přesností. Dobu výpočtu zkrátíme redukcí počtu násobiček. Méně zpožďovacích linek zase ušetří paměť. Na druhou stranu není vhodné z hlediska konečné délky slova využívat struktury co nejvíce minimalizované. Naopak složitější struktura (algoritmus s větším počtem operací) může být odolnější a zajistit si tak větší rezistenci před parazitní oscilací.

Výčet nejběžnějších způsobů realizace pro filtr typu IIR:

1. přímá forma (I. nekanonická a II. kanonická)
2. kaskádní forma (III. kanonická, sériová)
3. paralelní forma (IV. kanonická, využívá rozvoj na částečné zlomky)
4. ostatní formy:
    - duální struktura
    - křížová struktura
    - stavová struktura

Velmi se doporučuje 2. varianta, tedy vyjadřovat přenosovou funkci ve tvaru součinu dílčích subsystémů tzv. bikvadů (součiny racionálních lomených funkcí druhého řádu), které mohou být reprezentovány dvojicí reálných kořenových činitelů nebo komplexně sdružených pólů (nerozložitelné kvadratické trojčleny). Výhodou této realizace, obzvláště pro systémy vyšších řádů, je menší citlivost na nepřesnosti koeficientů způsobené kvantovacími vlivy při implementaci v reálných procesorech. viz kap 5.3.3.

Hlavní výhody:

1. bohatší možnosti realizace požadované přenosové funkce;
2. výrazně nižší řád přenosové funkce při stejných požadavcích na toleranční schéma, (až 10x nebo i více, nižší oproti FIR realizaci), z toho plynou nižší požadavky na výpočetní výkon systému;
3. méně náročné požadavky na paměť pro ukládání koeficientů přenosové funkce a menší zpoždění při zpracování vstupního vzorku;
4. poměrně jednoduchá metoda návrhu využívající podobnosti s analogovými systémy;
5. lze se přiblížit lineární charakteristice či považovat ji za lineární, ale pouze v úzkém rozsahu kmitočtů;
6. pokud máme všechna data k dispozici před filtrací, lze využít zpracování signálu „off-line“ a můžeme eliminovat nelineární fázové zkreslení;
7. pro určité pásmo kmitočtů lze vhodným rozmístěním dvojic pól-nula (pól p_i, nulový bod n_i v místě $1/p_i$) zajistit jednotkový přenos v amplitudové frekvenční charakteristice a aproximovat požadovaný tvar fázové charakteristiky-fázový korektor;
8. k filtru lze najít analogový ekvivalent.

Nevýhody:

1. nelze realizovat filtr s lineární fázovou frekvenční charakteristikou v celém kmitočtovém rozsahu;
2. obtížně využitelné pro adaptivní filtraci;

3. v případě vysokého řádu filtru a přítomnosti zpětnovazební smyčky platí zvýšená náchylnost na saturaci aritmetiky procesoru, velká citlivost na kvantování hodnot a na nepřesnosti způsobené zaokrouhlováním;
4. nevhodný návrh může vést k nestabilitě systému, a proto nutná zpětná kontrola, zda všechny póly výsledné přenosové funkce leží uvnitř jednotkové kružnice v rovině *z*.

3.2 Metody návrhu číslicových filtrů IIR:

Metody návrhu můžeme rozdělit takto:

3.2.1 Analogově číslicová transformace - návrh analogového prototypu

Jedná se o tradiční způsob návrhu číslicového filtru založený na podobnosti s analogovými systémy. Využívá aproximační úlohy pro aproximaci modulu normované ideální propusti (NDP), kterou pak transformujeme na požadovaný typ filtru (tzv. zpětná kmitočtová transformace) z přenosové funkce NDP neboli *H(p)* na přenosovou funkci analogového prototypu *H(s)*. Číslicový filtr vznikne převodem analogového prototypu přenosové funkce *H(s)* z s-roviny spojitého systému do z-roviny diskrétního systému. Touto operací získáme výslednou přenosovou funkci *H(z)*.

Použitá transformační rovnice musí zobrazovat s-rovinu do z-roviny tak, aby levá polorovina s-roviny odpovídala vnitřku jednotkové kružnice v z-rovině, případně jeho části a imaginární osa odpovídala jednotkové kružnici. Takto máme zaručeno, že se stabilní analogový filtr převede na stabilní číslicový filtr a racionálně lomená funkce *H(s)* rovněž na racionálně lomenou funkci *H(z)*.

Stručný přehled vlastností funkcí aproximujících ideální požadovanou kmitočtovou charakteristiku NDP:

Butterworthova aproximace:

- analogový prototyp má pouze póly,
- modulová kmitočtová charakteristika je monotónně klesající funkce,

- fázová charakteristika se nejvíce blíží k lineárnímu průběhu,
- strmost přechodového pásma je nejmenší.

Čebyševova aproximace I. typu:

- analogový prototyp má pouze póly,
- modulová kmitočtová charakteristika se zvlní v propustném pásmu.

Čebyševova aproximace II. typu:

- analogový prototyp má póly i nulové body,
- modulová kmitočtová charakteristika se zvlní v nepropustném pásmu.

Cauerova aproximace (podrobněji pojednáno v kap. 3.3):

- analogový prototyp má póly i nulové body,
- modulová kmitočtová charakteristika se zvlní v propustném i nepropustném pásmu,
- fázová kmitočtová charakteristika se nejvíce odlišuje od lineárního průběhu,
- strmost přechodového pásma je největší.

Základní metody pro uvedenou transformaci jsou:

- ➢ metoda náhrady diferenciální rovnice diferenční rovnicí
- ➢ metoda bilineámí transformace
- ➢ metoda invariantní impulsní odezvy

Metoda náhrady diferenciální rovnice diferenční rovnicí

Metoda využívá aproximace derivace funkce konečnou diferencí, ale je účinná až pro dostatečně vysoký vzorkovací kmitočet a pouze pro transformace dolních propustí na dolní propusti. Není příliš používaná a ani zmiňovaná v literatuře, protože se běžně využívá velmi oblíbená bilineární transformace.

Bilineární transformace

Provede se substituce argumentu s přenosové funkce *H(s)* analogového prototypu za argument z podle vztahu:

$$s = \frac{2}{T_{vz}}\left(\frac{1-z^{-1}}{1+z^{-1}}\right). \tag{3.4}$$

Transformace frekvencí z analogové do číslicové oblasti je ale nelineární. Nekonečná analogová kmitočtová osa se transformuje na konečný úsek číslicové kmitočtové osy. To znamená, že během návrhu musíme přepočítat mezní frekvence filtru, tzn. musíme provést tzv. předkreslení kmitočtové osy pro všechny kmitočty do analogové oblasti:

$$\omega = \frac{2}{T_{vz}} arctg \frac{\omega_a T}{2}, \ \omega_a = \frac{2}{T_{vz}} tg \frac{\omega T}{2}. \tag{3.5}$$

Důležité vlastnosti této metody jsou:

1. Stabilní filtr je transformován na stabilní číslicový filtr;
2. kmitočtové vlastnosti jsou zachované, např. původní analogový filtr typu dolní propusti musí být po transformaci číslicová dolní propust;
3. může dojít ke zlepšení přenosu v přechodových pásmech z důvodu zkreslení kmitočtové osy;
4. nezachovává časové vlastnosti
5. vhodná k návrhu filtrů s po částech konstantní kmitočtovou modulovou charakteristikou jako mají kmitočtově-selektivní filtry

V prostředí MATLAB můžeme k provedení bilineární transformace využít užitečný příkaz *Bilinear(bs,as,fvz,fp)*. Popis parametrů: *bs*, *as* jsou koeficienty analogové přenosové funkce *H(s)*, *fvz* vzorkovací kmitočet v [Hz] a *fp* je kmitočet okraje pásma v [Hz], které je při předkreslení identické s polohou okraje pásma analogového filtru.

Invariantní impulsní odezva

U této metody požadujeme zachování časových vlastností analogového prototypu. K impulsní odezvě známého analogového filtru hledáme impulsní odezvu číslicového filtru, který se shoduje ve vzorkovací periodě.

Navzorkujeme impulsní charakteristiku analogového filtru:

$$h[nT] = h_a(nT), n = 0,1,2,..., \tag{3.6}$$

analogový prototyp máme ve tvaru racionálně lomené funkce:

$$H_a(p) = \sum_{i=1}^{s} \frac{A_i}{p - p_i}, \tag{3.7}$$

impulsní charakteristika je pak dána:

$$h_a(t) = \sum_{i=1}^{s} A_i e^{p_i t} \tag{3.8}$$

po navzorkování dostaneme impulsní charakteristiku číslicového filtru:

$$h[nT] = h_a(nT) = \sum_{i=1}^{s} A_i e^{p_i nT}, \tag{3.9}$$

a přenosovou funkci:

$$H(z)=\sum_{i=1}^{s} A_i \frac{z}{z-e^{p_i T}} \qquad (3.10)$$

Důležité vlastnosti této metody jsou:

1. Zachovává časové vlastnosti;
2. lze zobecnit na jakoukoliv časovou charakteristiku analogového filtru (metoda tzv. signálové invariance);
3. kmitočtová osa zůstane zachována,
4. pouze u filtrů typu dolní nebo pásmová propust nedojde k překrývání spekter a kmitočtové vlastnosti zůstanou zachované.

Opět můžeme danou metodu realizovat v prostředí MATLAB funkcí *Impinvar(bs,as,fvz)*, vstupní parametry jsou totožné jako u funkce *Bilinear*

3.2.2 Iterační optimalizační postupy

Předchozí metody realizace pomocí analogového prototypu jsou limitovány z hlediska fyzikálních zákonů. Číslicové filtry mají ale mnohem větší volnost oproti analogovým protějškům, průběh jejich kmitočtové charakteristiky může být téměř libovolný. Tyto metody pro složitější a propracovanější návrh plně využívají předností výpočetních systémů a návrh často probíhá přímo v rovině *z*.

Metoda nejmenších čtverců

základní vlastnosti:

- můžeme navrhnout uživatelsky definovanou kmitočtovou charakteristiku

- nalezení koeficientů filtrů s požadovanou kmitočtovou charakteristikou převedeme na problém řešení lineárních a nelineárních rovnic.
- řešení těchto soustav je značně složité a výpočetně velmi náročné
- nemusíme získat stabilní filtr

Kmitočtová transformace

- spočívá v transformaci digitální dolní propusti na libovolný jiný typ číslicového filtru (dolní, horní, pásmová propust a pásmová zádrž)

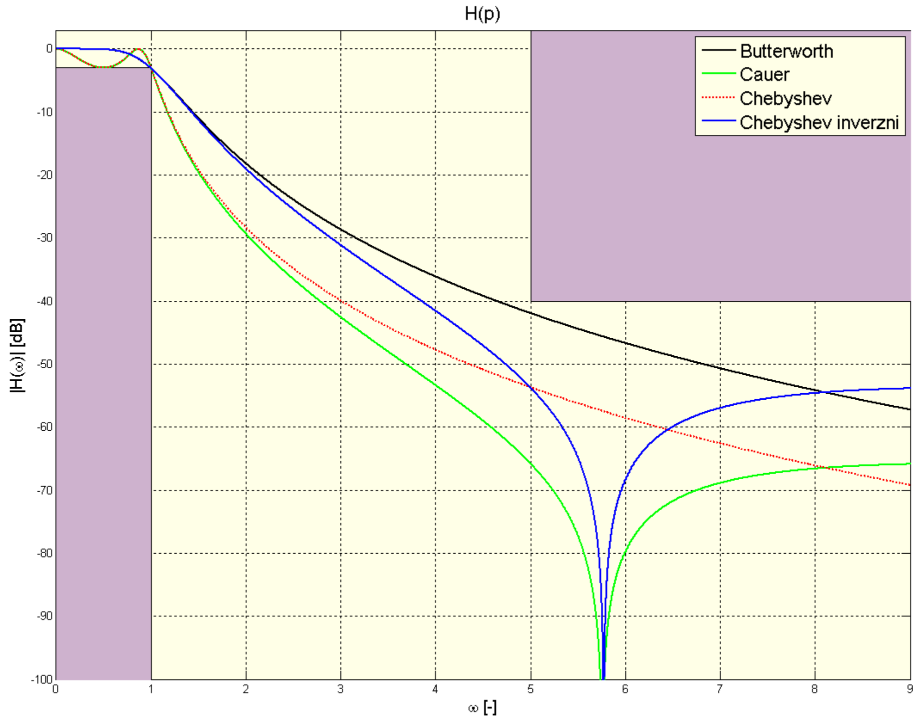
Analytické metody návrhu

- stejnoměrná aproximace číslicových filtrů IIR v rovině z.

3.3 Cauerova eliptická aproximace

Cauerovy eliptické filtry, známé také jako Zolotarevovy, se řadí mezi skupinu aproximací v Čebyševově smyslu. Mají izoextremální útlumovou charakteristiku podobnou jako u typu FIR se stejnoměrným zvlněním viz kap 4.3.3. Stejnoměrné zvlnění této aproximace se nachází jak v propustném, tak v nepropustném pásmu.

Stupeň aproximace (řád filtru) n společně s polohou zejména prvotních nul v přenosové funkci ovlivňují strmost modulové charakteristiky. Eliptické filtry se považují za optimální, a rovněž z hlediska řádu se vyznačují nejostřejším přechodovým pásmem, respektive nejnižším stupněm aproximace pro dané útlumové požadavky. Uvedené vlastnosti jsou názorně vyobrazeny na obr. 3.1.



Obr. 3.1 Modulová charakteristika NDP základních aproximací.

Mezi negativní vlastnosti patří *z* hlediska linearity v propustném pásmu v porovnání s ostatními aproximacemi (Čebyševovy, Butterworthova, Besselova,…) velmi špatná fázová charakteristika a dále pak velmi složitý návrh koeficientů filtru obtížně realizovatelný obyčejným matematickým aparátem či vědeckou kalkulačkou se základními funkcemi.

Veškerou aproximační úlohu vztahujeme k normování ke kmitočtu okraje propustného pásma.

$$\Omega = \frac{\omega}{\omega_p} = \frac{\widetilde{\Omega}}{\sqrt{k}}, \tag{3.11}$$

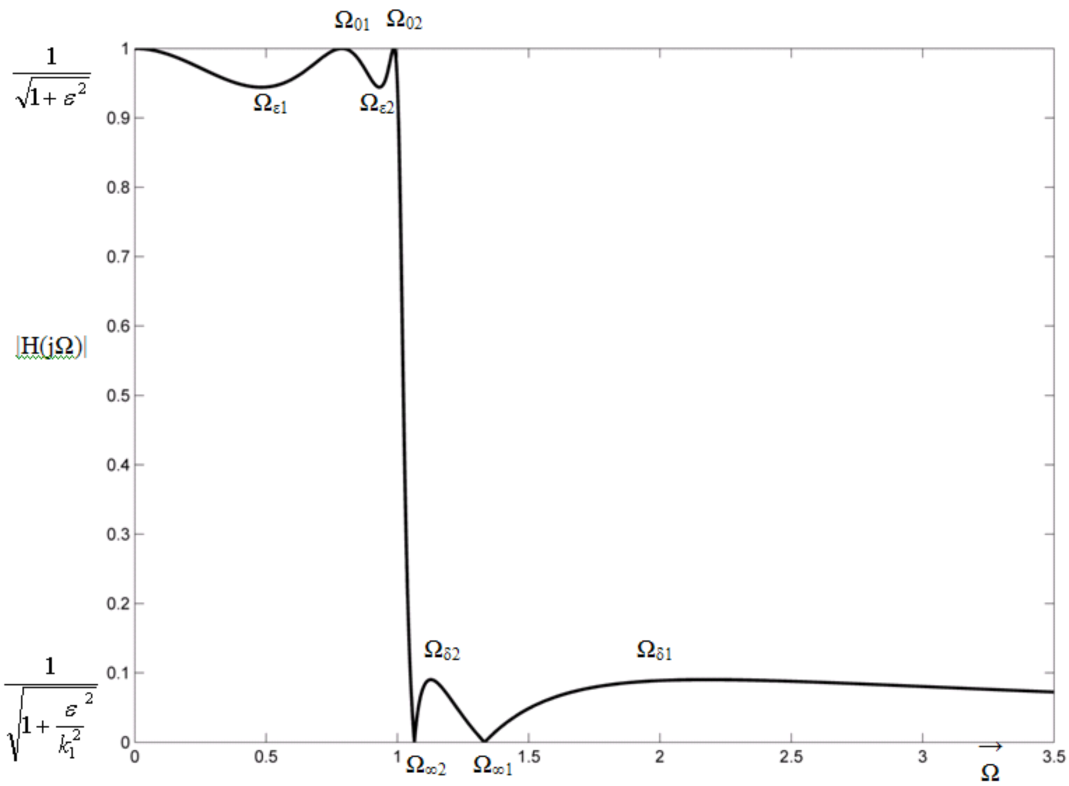
$$\Omega_p = 1,$$

$$\Omega_s = \frac{1}{k}.$$

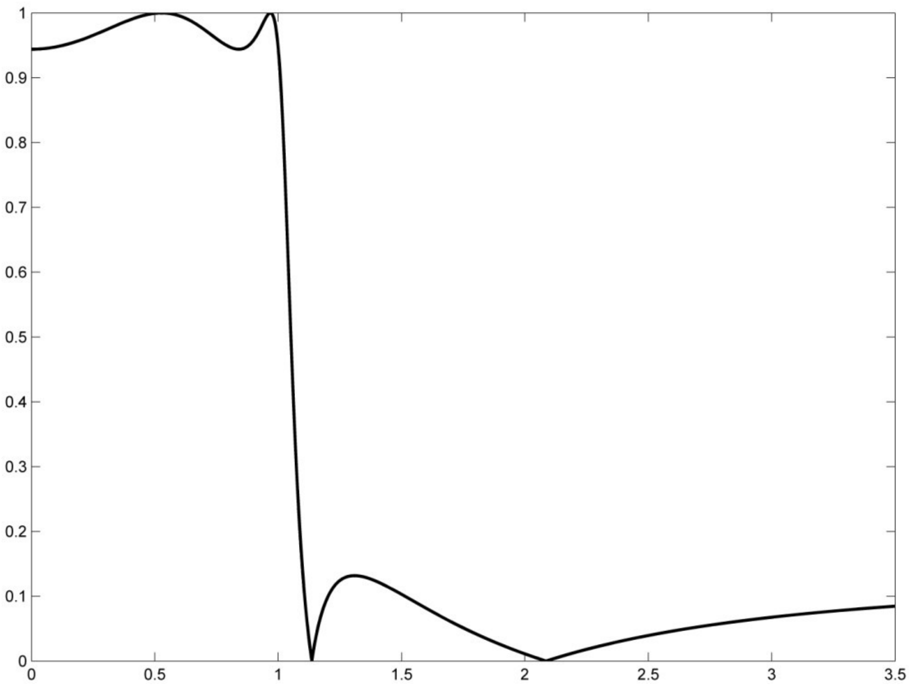
Existuje ještě hojně využívaná varianta normování ke geometrickému středu frekvencí propustného a závěrného (neboli nepropustného) pásma se symetrií pro nulové body a póly přenosové funkce ve výsledném vztahu nebo normování k okraji nepropustného pásma.

Charakteristika vykazuje proměnný počet lokálních extrémů podle řádu systému.

V propustném pásmu dosahuje maximální hodnoty útlumu v normovaných kmitočtech $\Omega_{\varepsilon\mu}$, nulový útlum naopak v $\Omega_{0\mu}$ a v případě lichého stupně aproximace navíc ještě v počátku Ω=0. V nepropustném pásmu dostaneme maximální útlum v $\Omega_{\infty\mu}$, popřípadě minimální v $\Omega_{\delta\mu}$. Charakteristika Cauerova filtru je znázorněna na obrázku 3.2.



Obr. 3.2 Průběh modulové charakteristiky Cauerova filtru 5.řádu



Obr. 3.3 Ukázka průběhu modulové charakteristiky Cauerova filtru 4.řádu

Charakteristická rovnice:

$$H(j\Omega)H(-j\Omega) = \frac{1}{1+\varepsilon^2\varphi^2(\Omega)} = Q(\Omega^2) \tag{3.12}$$

rozlišuje lichý a sudý stupeň aproximace pro charakteristickou funkci. Ta má tvar racionálně lomené funkce proměnné Ω

pro lichý stupeň *n=2m+1*:

$$\varphi(\Omega) = C\Omega\prod_{\mu=1}^{m}\frac{\Omega^2-\Omega_{0\mu}^2}{1-k^2\Omega_{0\mu}^2\Omega^2} \tag{3.13}$$

pro sudý stupeň $n=2m$:

$$\varphi(\Omega) = C\prod_{\mu=1}^{m}\frac{\Omega^2 - \Omega_{0\mu}^2}{1 - k^2\Omega_{0\mu}^2\Omega^2}. \tag{3.14}$$

$\Omega_{0\mu}$, $\Omega_{\infty\mu}$ jsou nulové body a póly, jsou navzájem reciproké a splňují tedy: $\Omega_{0\mu}\Omega_{\infty\mu} = 1/k$. Stanovení násobné konstanty C řeší poměrně složitá aproximační úloha. Charakteristická funkce $\varphi(\Omega)$ odpovídá diferenciální rovnici eliptického typu a spolu s průběhem modulové charakteristiky musí splňovat tyto následující podmínky:

1. v počátku $\Omega=0$ je $\varphi(\Omega)=0$ nebo $\varphi(\Omega)=1$;
2. $\Omega_{\infty\mu} = 1/k\Omega_{0\mu}$;
3. $\varphi(\Omega)\rightarrow\infty$ pro všechna $\Omega = \pm\Omega_{\infty\mu}$, pro $\mu=1,2,\ldots,m$;
4. $\varphi(\Omega)=0$ pro všechna $\Omega = \pm\Omega_{0\mu}$, pro $\mu=1,2,\ldots,m$;
5. $\pm\Omega_{\varepsilon\mu}$ jsou lokální extrémy charakteristické funkce v propustném pásmu po jeho okraj v normovaném kmitočtu $\Omega=\Omega_p$, v těchto bodech $\varphi(\Omega)=\pm1$ alternuje právě v počtu m;
6. $\pm\Omega_{\delta\mu}$ jsou analogicky lokální extrémy charakteristické funkce v nepropustném pásmu od jeho okraje v $\Omega=\Omega_s$ po $\Omega=\infty$, ve kterých $\varphi(\Omega)=\pm 1/k_1$ alternuje také v počtu m;
7. $\Omega_{\delta\mu} = 1/k\Omega_{\varepsilon\mu}$;
8. derivace modulu přenosové funkce je nulová právě v lokálních extrémech propustného a nepropustného pásma s výjimkou jejich okrajů a platí:

$$\frac{dQ(\Omega^2)}{d\Omega} = 0 \text{ pro } \Omega = \pm\Omega_{\varepsilon\mu} \text{ a pro } \Omega = \pm\Omega_{\delta\mu}. \tag{3.15}$$

Primární parametry obvyklého tolerančního schématu představují určitý zaběhnutý standard vyjádření požadavků na řešení aproximační úlohy syntézy NDP. Mezi ně zahrnujeme přípustné chyby aproximace v propustném a nepropustném pásmu dané parametry a_p a a_s v [dB], a vymezení přechodového pásma pomocí normovaných hraničních frekvencí Ω_p, Ω_s. Z primárních lze snadno vyjádřit parametry sekundární:

$$k_1 = \sqrt{\frac{10^{0.1ap}-1}{10^{0.1as}-1}}, \tag{3.16}$$

$$\varepsilon = \sqrt{10^{0.1ap}-1}, \tag{3.17}$$

$$\kappa = \frac{1}{\Omega_s}, \tag{3.18}$$

$$\kappa^2 + \kappa'^2 = 1, \qquad k_1^2 + k_1'^2 = 1. \tag{3.19}$$

Stupeň aproximace je jak v lichém, tak v sudém případě totožný, musí být kladné celé číslo, vždy tedy zaokrouhlujeme nahoru k nejbližší vyšší číslici, určíme přímo z rovnice:

$$n \geq \frac{K(\kappa'_1)}{K(\kappa_1)} \frac{K(\kappa)}{K(\kappa')}. \tag{3.20}$$

K(κ) je úplný eliptický integrál I. druhu, známý také jako velikost čtvrtperiody. *κ'* je doplňkový modul odvozený od modulu κ dle vztahu 3.17. Bližší popis viz kap. 3.3.4.

3.3.1 Lichý stupeň aproximace

platí *n=2m+1*, charakteristická funkce má tento tvar:

$$\varphi(\Omega) = (-1)^m \sqrt{\frac{k^n}{k_1}}\Omega \prod_{\mu=1}^{m} \frac{\Omega^2 - \Omega_{0\mu}^2}{1 - k^2\Omega_{0\mu}^2\Omega^2}\,. \tag{3.21}$$

Nulové body charakteristické funkce nalezneme v těchto bodech a to podle vztahu:

$$\Omega_{0\mu} = sn\left(\frac{2\mu}{n}K(\kappa)\,|\,\kappa\right), \mu = 0,1,...,m\,. \tag{3.22}$$

Póly charakteristické funkce jsou v těchto bodech a určíme je podle vztahu:

$$\Omega_{\infty\mu} = \frac{1}{\kappa\Omega_{0\mu}}\,. \tag{3.23}$$

Body, kde má frekvenční charakteristika předepsanou hodnotu útlumu a_p v propustném pásmu:

$$\Omega_{\varepsilon\mu} = sn\left(\frac{2\mu+1}{n}K(\kappa)\,|\,\kappa\right), \mu = 0,1,...,m\,. \tag{3.24}$$

Póly charakteristické funkce představují zároveň nulové body přenosové funkce *H(p)*, které leží vždy na imaginární ose viz obr.3.5 ukázka rozložení nulových bodů a pólů Cauerova filtru lichého řádu

Protože jsme zaokrouhlili stupeň aproximace (původně kladné reálné číslo), musíme provést korekci diskriminačního koeficientu κ_1 na κ_{1n}, a proto přepočteme specifikace v tomto smyslu:

$$k_{1n} = \kappa^n \prod_{\mu=1}^{m} \left[\frac{1-\Omega_{0\mu}^2}{1-\kappa^2\Omega_{0\mu}^2} \right]^2 . \tag{3.25}$$

Získáme nový odstup modulu přenosu v propustném a nepropustném pásmu k_{1n}. Tím pádem dojde ke změně útlumu v nepropustném pásmu a_s.

$$a_s = 10 \log\left(1 + \frac{\varepsilon^2}{k_{1n}^2} \right) \tag{3.26}$$

Než stanovíme záporný reálný pól α_0 (3.27), připravíme tyto pomocné parametry θ a v_∞, a je nezbytné v následujících vztazích vzít v úvahu dříve zkorigovaný koeficient κ_{1n}:

$$\theta = \arctan\left(\frac{1}{\varepsilon}\right), \qquad v_\infty = \frac{K(\kappa)}{nK(\kappa_{1n})} F(\theta \,|\, \kappa'_{1n}). \tag{3.27}$$

$F(\theta \,|\, k'_1)$ je neúplný eliptický integrál I.druhu (kap 3.3.4). Výpočet argumentu $\boldsymbol{v_\infty}$ si lze usnadnit zavedením přibližného vztahu:

$$v_\infty \approx \frac{2K(k)}{\pi n} \ln\left(\frac{1}{\varepsilon} + \sqrt{1+\frac{1}{\varepsilon^2}} \right) \tag{3.28}$$

nebo nahrazením eliptických funkcí nekonečnou řadou, avšak tento algoritmus může velmi znehodnotit přesnost vyčíslení koeficientů přenosové funkce.

$$\alpha_0 = -sc(v_\infty \mid \kappa') \tag{3.29}$$

Podíl eliptických Legenderových (Jacobiho) funkcí sinus a kosinus naznačuje způsob výpočtu odvozené funkce $sc(v_\infty \mid \kappa') = sn(v_\infty \mid \kappa') / cn(v_\infty \mid \kappa')$.(viz kap 3.3.4)

Póly přenosové funkce $\alpha_\mu \pm j\beta_\mu$ vyjádříme za pomoci k, α_0 a $\Omega_{0\mu}$:

$$\alpha_\mu = \alpha_0 \frac{\sqrt{1-\Omega_{0\mu}^2}\sqrt{1-\kappa^2\Omega_{0\mu}^2}}{1+\kappa^2\Omega_{0\mu}^2\alpha_0^2} \tag{3.30}$$

$$\beta_\mu = \Omega_{0\mu} \frac{\sqrt{1+\alpha_0^2}\sqrt{1+\kappa^2\alpha_0^2}}{1+\kappa^2\Omega_{0\mu}^2\alpha_0^2}.$$

V levé polorovině komplexního kmitočtu dostáváme komplexně sdružené póly.

Počet pólů odpovídá *n*, počet nul je *n-1*.

Přenosovou funkci *H(p)* ve tvaru podílu čitatele a jmenovatele sestavíme pomocí normovací konstanty (násobná konstanta) H_0 a pólů.

$$H_0 = -\alpha_0 \prod_{\mu=1}^{m} \frac{\alpha_\mu^2 + \beta_\mu^2}{\Omega_{\infty\mu}^2} \tag{3.31}$$

$$H(p)=\frac{H_0}{p-\alpha_0}\prod_{\mu=1}^{m}\frac{p^2+\Omega_{\infty\mu}^2}{p^2-2\alpha_\mu p+\alpha_\mu^2+\beta_\mu^2} \tag{3.32}$$

3.3.2 Sudý stupeň aproximace

platí *n=2m*, charakteristická funkce má tento tvar:

$$\varphi(\Omega)=(-1)^m\sqrt{\frac{\kappa^n}{\kappa_1}}\prod_{\mu=1}^{m}\frac{\Omega^2-\Omega_{0\mu}^2}{1-\kappa^2\Omega_{0\mu}^2\Omega^2}\,. \tag{3.33}$$

Nulové body charakteristické funkce nalezneme v těchto bodech a to podle vztahu:

$$\Omega_{0\mu}=sn\left(\frac{2\mu-1}{n}K(\kappa)\,|\,\kappa\right),\mu=1,\dots,m\,. \tag{3.34}$$

Protože jsme zaokrouhlili stupeň aproximace (původně kladné reálné číslo), musíme provést korekci diskriminačního koeficientu κ_1 na κ_{1n}, a proto přepočteme specifikace v tomto smyslu:

$$k_{1n}=\kappa^n\prod_{\mu=1}^{m}\Omega_{0\mu}^4\,. \tag{3.35}$$

Získáme nový odstup modulu přenosu v propustném a nepropustném pásmu κ_{1n}. Tím pádem dojde ke změně útlumu v nepropustném pásmu a_s.viz (3.24)

Body, kde má frekvenční charakteristika předepsanou hodnotu útlumu a_p v propustném pásmu jsou:

$$\Omega_{\varepsilon\mu} = sn\left(\frac{2\mu}{n}K(\kappa) \mid \kappa\right), \mu = 1,\dots,m\,. \tag{3.36}$$

Póly charakteristické funkce (zároveň nulové body přenosové funkce), pomocné parametry a póly přenosové funkce určíme shodně jako u lichého řádu viz vztahy (3.21).

Ukázka rozložení nulových bodů a pólů Cauerova filtru sudého řádu je na obr 3.6.

Výsledná přenosová funkce *H(p)* má tedy tvar:

$$H_0 = \frac{1}{\sqrt{1+\varepsilon^2}} \prod_{\mu=1}^{m} \frac{\alpha_\mu^2 + \beta_\mu^2}{\Omega_{\infty\mu}^2} \tag{3.37}$$

$$H(p) = H_0 \prod_{\mu=1}^{m} \frac{p^2 + \Omega_{\infty\mu}^2}{p^2 - 2\alpha_\mu p + \alpha_\mu^2 + \beta_\mu^2} \tag{3.38}$$

Sudý řád má shodný počet nulových bodů a pólů v přenosové funkci.

Uvedené vzorce platí pouze pro výpočet sudého řádu aproximace pro případ typu A, který je realizovatelný pouze s vázanými induktory. Přenosová funkce má v tomto případě konečný útlum v nekonečném normovaném kmitočtu **Ω**.

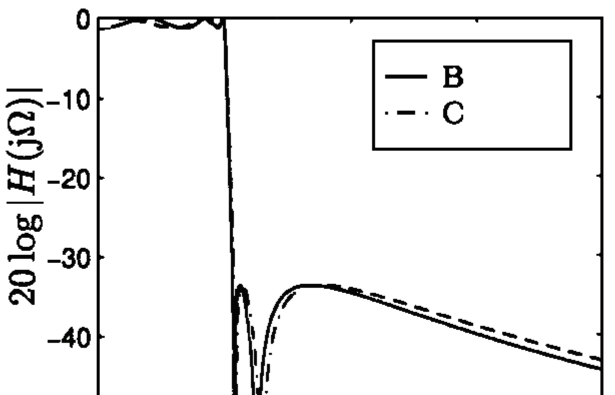
Pokud požadujeme strukturu bez vázaných induktorů, využijeme další variantu řešení aproximace, typ B spojený s touto transformací:

$$s_B^2 = \frac{\left(\Omega_{\infty 1}^2 - 1\right)s^2}{\Omega_{\infty 1}^2 + s^2}. \tag{3.39}$$

Z případu B lze odvodit ještě třetí variantu, typ C, kde transformujeme nejnižší nulový bod charakteristické funkce do počátku vztahem:

$$s_C^2 = \frac{s_B^2 + \Omega_{01B}^2}{1 - \Omega_{01B}^2}. \tag{3.40}$$

Výsledná přenosová funkce bude podobná výsledku v lichém řádu a lze ji tedy realizovat strukturou LC s totožnými zakončovacími odpory. Uvedené případy řešení sudého řádu vystihuje obr. 3.3 a obr. 3.4. Obr. 3.4 je převzat z [3].



Obr. 3.4 Možné případy řešení sudého řádu pro typy B a C

3.3.3 Zdrojový kód pro návrh koeficientů NDP Caurovy aproximace:

```
function [cit,jmen,n,z,p,k,alfa0]=cauer(omegas,ap,as,n)
  kappa= omegas^-1; % (3.18)
    if n<1
          kappa1=((10^(.1 * abs(ap)) - 1)  /  (10^(.1 * abs(as)) - 1))^.5;   % (3.16)
%%%%%      vypocet radu (3.20)
             n  =  ceil(uelint(kappa)  *  uelint((1  -  kappa1^2)^.5)  /  (uelint((1  -  kappa^2)^.5)  *
uelint(kappa1)));
    end;
%%%%%%%%%%%%%%%%%%
if floor(n/2) * 2 ==n      % sudy rad
   m=n * .5;
%%%%%%   vypocet omega0mi, nuly charakt. fce, mi=1, 2, ...m (3.34)
         for mi=1:m, %  zaciname az primo od omega01
              jac=jacob((uelint(kappa) * (2 * mi-1)) / n, kappa);
              omega0mi(mi)=jac(1);
   %         poly charakt. fce (3.23)
              omegainfmi(mi)=(kappa * omega0mi(mi))^-1;
         end;
 %%%%%  prepocet kappa1.... misto kappa1...kappa1n (3.25)
    prep=1;
        for mi=1:m,
             prep=prep * omega0mi(mi)^4;
        end;
    kappa1n=kappa^n * prep;
%%%     epsilon=(10^(.1 * abs(ap)) - 1)^.5;  (3.17)
% % %    asnew=10  *  log10(1  +  (epsilon  /  kappa1n)^2)       % novy  utlum  v  zav.  pasmu  dle
"noveho" radu (3.26)
```

```
vinf=(nelint(atan((10^(.1 * abs(ap)) - 1)^-.5),(1 - kappa1n^2)^.5) / n) * (uelint(kappa) / uelint(kappa1n));  % pomocny par. (3.27)

        jac=jacob(vinf,(1 - kappa^2)^.5);      %ellipt. sin & cos

    % % % % %  realny pol alfa0 (3.29)

     alfa0= -jac(1) / jac(2);

      % poly prenosove fce alfa beta (3.30)

        for mi=1:m,

            jmen(mi)=1 + (kappa * omega0mi(mi) * alfa0)^2;

              alfa(mi)= alfa0 * ((1 - omega0mi(mi)^2) * (1 - (kappa * omega0mi(mi))^2))^.5 / jmen(mi);

              beta(mi)= omega0mi(mi) * (((1 + alfa0^2) * (1 + (kappa * alfa0)^2))^.5) / jmen(mi);

        end;

    % % % % %  urceni H0 (3.37)

       H0=((1 + (10^(.1 * abs(ap)) - 1))^-.5);

            for mi=1:m,

                H0=H0 * (alfa(mi)^2 + beta(mi)^2) / (omegainfmi(mi)^2);

            end;

     % citatel,  jmenovatel (3.38)

       cit=H0;

       jmen=1;

           for mi=1:m,

               cit=conv(cit,[1 0 omegainfmi(mi)^2]);

               jmen=conv(jmen,[1 -2*alfa(mi) alfa(mi)^2+beta(mi)^2]);

           end;

    % % % % % % % % % % % % % % % % %

    else                        % lichy rad

      m=(n-1)  * .5;

       % % % % % %   vypocet omega0mi, nuly charakt. fce (3.22)
```

```
for mi=1:m,
    jac=jacob((uelint(kappa)*2*mi) / n, kappa);
    omega0mi(mi)=jac(1);
%       poly charakt. fce (x.y)
    omegainfmi(mi)=(kappa * omega0mi(mi))^-1;
end;
% % % % %  prepocet kappa1.... misto kappa1...kappa1n (3.25)
prep=1;
for mi=1:m,
    prep=prep * (1 - omega0mi(mi)^2) / (1 - (kappa*omega0mi(mi))^2) * (1 - omega0mi(mi)^2) / (1 - (kappa*omega0mi(mi))^2);
end;
kappa1n=kappa^n * prep;
%    epsilon=(10^(.1 * abs(ap)) - 1)^.5;
% % %    asnew=10 * log10(1 + (epsilon / kappa1n)^2)      % novy utlum v zav. pasmu dle "noveho" radu
    vinf=(nelint(atan((10^(.1 * abs(ap)) - 1)^-.5),(1 - kappa1n^2)^.5) / n) * (uelint(kappa) / uelint(kappa1n));
    jac=jacob(vinf,(1 - kappa^2)^.5);
alfa0= -jac(1) / jac(2);
    % poly prenosove fce alfa beta shodne
for mi=1:m,
    jmen(mi)=1 + (kappa * omega0mi(mi) * alfa0)^2;
    alfa(mi)= alfa0 * ((1 - omega0mi(mi)^2) * (1 - (kappa * omega0mi(mi))^2))^.5 / jmen(mi);
    beta(mi)= omega0mi(mi) * (((1 + alfa0^2) * (1 + (kappa * alfa0)^2))^.5) / jmen(mi);
end;
% % % % %  urceni H0 (3.31)
H0= -alfa0;
```

```
        for mi=1:m,
            H0=H0 * (alfa(mi)^2 + beta(mi)^2) / (omegainfmi(mi)^2);
        end;
  % citatel,  jmenovatel (3.32)
   cit=H0;
   jmen=[1 -alfa0];
        for mi=1:m,     %vysledek
            cit=conv(cit,[1 0 omegainfmi(mi)^2]);
            jmen=conv(jmen,[1 -2*alfa(mi) alfa(mi)^2+beta(mi)^2]);
        end;
end;
% % % % % % END% % % % % % END% % % % % % % %END % % % % % % % %
```

3.3.4 Eliptické funkce a integrály

Úplný eliptický integrál

K(κ) je úplný eliptický integrál I. druhu, známý také jako velikost čtvrtperiody. Modul úplného eliptického integrálu κ může nabývat libovolných hodnot z intervalu 0 až 1. Mezní hodnoty jsou K(0) = π/2 a K(1) = ∞.

Postup výpočtu:

$M(a_0,b_0)$ je společná limita aritmetického a geometrického průměru čísel a_0 a b_0.

1) Položíme-li $a_0 = 1, b_0 = \sqrt{1-\kappa^2} = \kappa'$, $c_0 = \kappa$ potom úplný eliptický integrál modulu κ je roven:

$$K(\kappa) = \frac{\pi}{2M(1,\kappa')} \,. \tag{3.41}$$

2) K zadanému modulu k sestavíme s předem danou přesností ε posloupnosti $\{a_m\}$, $\{b_m\}$ a $\{c_m\}$ dle výrazů:

$$a_{m+1} = \frac{a_m + b_m}{2} \,, \tag{3.42}$$

$$b_{m+1} = \sqrt{a_m b_m} \,, \tag{3.43}$$

$$c_{m+1} = \frac{a_m - b_m}{2} \,. \tag{3.44}$$

V prostředí MATLAB volíme toleranci např. *eps* nebo 10^{-9}.

3) Algoritmus ukončíme pro jistou hodnotu N, pro kterou platí $c_N \leq \varepsilon$. Získáme společnou limitu: $M(1,k') = a_N$ a určíme integrál:

$$K(k) = \frac{\pi}{2a_N} \,. \tag{3.45}$$

Záměnou $a_0 = 1$, $b_0 = \kappa$, $c_0 = \sqrt{1-\kappa^2} = \kappa'$ dostáváme eliptický integrál doplňkového modulu κ':

$$K(\kappa') = \frac{\pi}{2M(1,\kappa)}\,. \tag{3.46}$$

Zdrojový kód pro úplný eliptický integrál:

```
function vysl=uelint(k)
tol=eps; % tolerance
   a0 = 1;
   b0 = (1 - k^2)^.5;  % uvodni podminky, modul k preveden na kvadrat jako m parametr
      ind = 1;
      c1 = k;
  while (c1 >= tol)           % cN<epsilon
     ind = ind + 1;
         a1 = (a0+b0) * .5;
         b1 = (a0*b0)^.5;
         c1 = (a0-b0) * .5;
      a0 = a1;
      b0 = b1;
  end;
 vysl = pi / (2*a1);
% % % % % % END% % % % % END% % % % % % %END % % % % % % %
```

Neúplný eliptický integrál

F(φ|κ) je neúplný eliptický integrál I. druhu.

Postup výpočtu:

1) Využijeme posloupnosti $\{a_m\}$, $\{b_m\}$ a $\{c_m\}$ z výpočtu úplného eliptického integrálu.

2) K zadanému argumentu φ určíme vzestupně posloupnost $\{\phi_m\}$ podle rekurentního vztahu $\phi_0 = \phi$, $\phi_{m+1} = \phi_m + \arctan(\tan\phi_m b_m / a_m)$. Je nutné zaručit, aby $\{\phi_m\}$byla rostoucí posloupnost, která musí splňovat $(\phi_{m+1} \sim 2\phi_m)$. Potom platí:

$$F(\phi \mid \kappa) = \frac{\phi_N}{2^N a_N}. \tag{3.47}$$

Zdrojový kód pro neúplný eliptický integrál:

```
function vyst=nelint(fi,k)
 tol=eps;    % tolerance
   a0 = 1;
   b0 = (1 - k^2)^.5;
   fi0=fi;
     ind = 1;
     c1 = k;
   while (c1 >= tol)       % cN<epsilon
     ind = ind + 1;
        a1 = (a0 + b0) * .5;
        b1 = (a0 * b0)^.5;
        c1 = (a0 - b0) * .5;
          t=floor(.5 * (1 + 2 * fi0/pi));
          fi1=fi0 + atan2(b0 * tan(fi0),a0) + t*pi;
     a0 = a1;
     b0 = b1;
     fi0=fi1;
   end;
```

```
vyst=fi1 / (a1 * 2^(ind - 1)) ;

% % % % % % END% % % % % % END% % % % % % % %END % % % % % % % %
```

Jacobiho eliptické funkce

Jacobiho eliptické funkce jsou funkce inverzní k eliptickým integrálům, a tedy algoritmus jejich výpočtu je obdobný předcházejícím.

Postup výpočtu:

1) Opět využijeme posloupnosti $\{a_m\}$, $\{b_m\}$ a $\{c_m\}$ z předcházejících výpočtů.
2) K zadanému argumentu u určíme

$$\phi_N = 2^N a_N u \,. \tag{3.48}$$

3) Nalezneme sestupně posloupnost $\{\phi_m\}$ dle rekurentní formule:

$$\phi_{m-1} = \frac{1}{2}[\phi_m + \arcsin(\frac{c_m}{a_m}\sin\phi_m)]\,. \tag{3.49}$$

Eliptické funkce jsou potom určeny vztahy:

$$sn(u|\kappa) = \sin\phi_0\,, \tag{3.50}$$

$$cn(u|\kappa) = \cos\phi_0\,, \tag{3.51}$$

$$dn(u|\kappa) = \frac{\cos\phi_0}{\cos(\phi_1 - \phi_0)}\,. \tag{3.52}$$

V případě argumentu rovného lichým násobkům $K(\kappa)$ je $cos\,\phi_0 = 0$, a tím pádem funkci $dn(u|\kappa)$ získáme ze vztahu:

$$dn(\mathrm{K}(\kappa)|\kappa) = \kappa'. \tag{3.53}$$

Funkce $dn(u|\kappa)$ není zpracována ve zdrojovém kódu Jacob.m, protože není využita v návrhu Cauerových filtrů.

Zdrojový kód pro Jacobiho eliptické funkce:

```
function vyst=jacob(u,k)
 tol=eps; %presnost
   a0 = 1;
   b0 = (1 - k^2)^.5;
    ind = 1;
    c1 = 1;
  am(ind)=a0;
  cm(ind)=k;
 while (c1 >= tol)                  % cN<epsilon
    a1 = (a0 + b0) * .5;
    b1 = (a0 * b0)^.5;
    c1 = (a0 - b0) * .5;
       ind = ind + 1;
       a0 = a1;
       b0 = b1;
         am(ind)=a0;
         cm(ind)=c1;
 end;
```

```
N=ind-1; %primo index N
   fin=2^N * a1 * u; %posledni prvek ve fim
   fim(ind)=fin;
      while (ind~=1)    %po nulty prvek
         fi0=.5 * (fim(ind) + asin((cm(ind) / am(ind)) * sin(fim(ind))));
         fim(ind-1)=fi0;
         ind=ind - 1;
      end;
vyst=[sin(fim(ind)) cos(fim(ind))];
% % % % % % END% % % % % % END% % % % % % %END % % % % % % %
```

3.3.5 Příklad výpočtu koeficientů NDP pro 5.řád filtru

Zadání NDP:

omegas =   1; omegas =   1,05; ap =   0,5 dB; as =   15 dB;

Primární parametry:

kappa =

0,952 380 952 380 95

kappa1 =

0,063 123 413 574 66

epsilon =

0,349 311 400 188 95

Úplný eliptický integrál I.druhu:

$K(\kappa_1')$ =

4,152 101 367 962 32

$K(\kappa_1)$ =

1,572 364 578 748 43

$K(\kappa)$ =

1,609 347 820 651 64

$K(\kappa')$ =

2,61243911115466

Výpočet stupně aproximace:

n =

5

m =

2

Výpočet nulových bodů charakteristické funkce :

omega0mi =

0,788 396 155 268 83   0,986 264 659 579 23

Výpočet pólů charakteristické funkce :

omegainfmi =

±1,331 817 758 093 93i   ±1,064 622 958 758 31i

Přepočet specifikace:

kappa1n =

0,0316 737 896 901 2

Přepočet na nový útlum:

asnew =

20,885 816 075 322 20

Argument pro neúplný eliptický integrál:

phi =

1,234 735 091 020 09

Neúplný eliptický integrál I.druhu:

F =

1,772 408 531 865 95

Pomocný parametr:

vinf =

0,589 401 389 758 38

Záporný reálný pól:

alfa0 =

-0,664 419 730 928 85

Pólý přenosové funkce:

alfa =

-0,216 155 694 150 08 -0,027 098 431 544 70

beta =

0,896 914 103 716 15   1,008 479 365 208 57

Normovací konstanta:

H0 =

0,286 305 205 718 10

Koeficienty přenosové funkce H(p):

cit =

0,286 305 205 718 10

0

0,832 335 209 369 65

0

0,575 586 371 663 29

jmen =

1,000 000 000 000 00

1,150 927 982 318 42

2,215 618 751 208 73

1,743 452 574 503 38

1,189 288 762 194 70

0,575 586 371 663 29

nuly =

0 + 1,331 817 758 093 93i

0 - 1,331 817 758 093 93i

0 + 1,064 622 958 758 31i

0 - 1,064 622 958 758 31i

poly =

-0,0270 984 315 447 1 + 1,008 479 365 208 57i

-0,027 098 431 544 71 - 1,008 479 365 208 57i

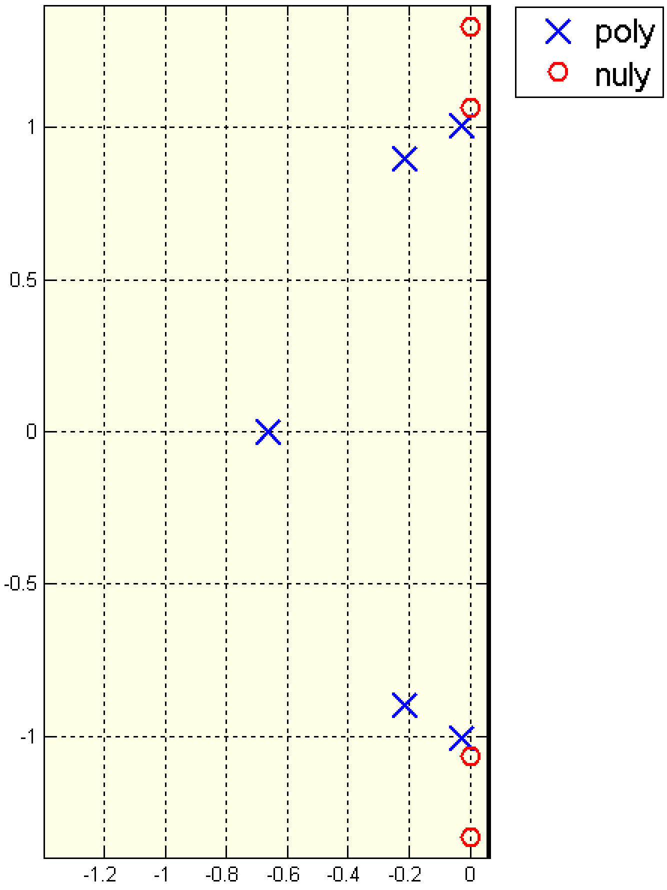
-0,216 155 694 150 08 + 0,896 914 103 716 15i

-0,216 155 694 150 08 - 0,896 914 103 716 15i

-0,664 419 730 928 85

K =

0,286 305 205 718 10



Obr. 3.5 Ukázka rozložení nulových bodů a pólů Cauerova filtru 5. řádu

Příklad výpočtu koeficientů NDP pro 4.řád filtru

Zadání NDP:

omegas = 1; omegas = 1,1; ap = 0,5 dB; as = 15 dB;

Primární parametry:

kappa =

0,909 090 909 090 91

kappa1 =

0,063 123 413 574 66

epsilon =

0,349 311 400 188 95

Úplný eliptický integrál I.druhu:

$K(\kappa_1')$ =

4,152 101 367 962 32

$K(\kappa_1)$ =

1,572 364 578 748 43

$K(\kappa)$ =

1,646 530 221 289 42

$K(\kappa')$ =

2,321 924 036 776 12

Výpočet stupně aproximace:

n =

4

m =

2

Výpočet nulových bodů charakteristické funkce :

omega0mi =

0,527 413 848 055 67   0,968 148 816 723 01

Výpočet pólů charakteristické funkce :

omegainfmi =

±2,085 648 687 563 26i   ±1,136 188 962 894 45i

Přepočet specifikace:

kappa1n =

0,046 430 552 335 28

Přepočet na nový útlum:

asnew =

17,604 238 472 213 52

Argument pro neúplný eliptický integrál:

phi =

1,234 735 091 020 09

Neúplný eliptický integrál I.druhu:

F =

1,770 438 861 703 55

Pomocný parametr:

vinf =

0,653 905 197 515 19

Záporný reálný pól:

alfa0 =

-0,754 695 968 065 56

Pólý přenosové funkce:

alfa =

-0,497 541 472 669 61   -0,062 241 131 555 15

beta =

0,708 544 851 892 26   1,020 633 337 711 78

Normovací konstanta:

H0 =

0,131 761 362 290 97

Koeficienty přenosové funkce H(p):

cit =

0,131 761 362 290 97

0

0,743 246 845 661 71

0

0,739 897 434 878 60

jmen =

1,000 000 000 000 00

1,119 565 208 449 53

1,919 019 869 693 51

1,133 735 090 102 87

0,783 739 114 144 28

nuly =

0 + 2,085 648 687 563 26i

0 - 2,085 648 687 563 26i

0 + 1,136 188 962 894 45i

0 - 1,136 188 962 894 45i

poly =

-0,062 241 131 555 15 + 1,020 633 337 711 78i

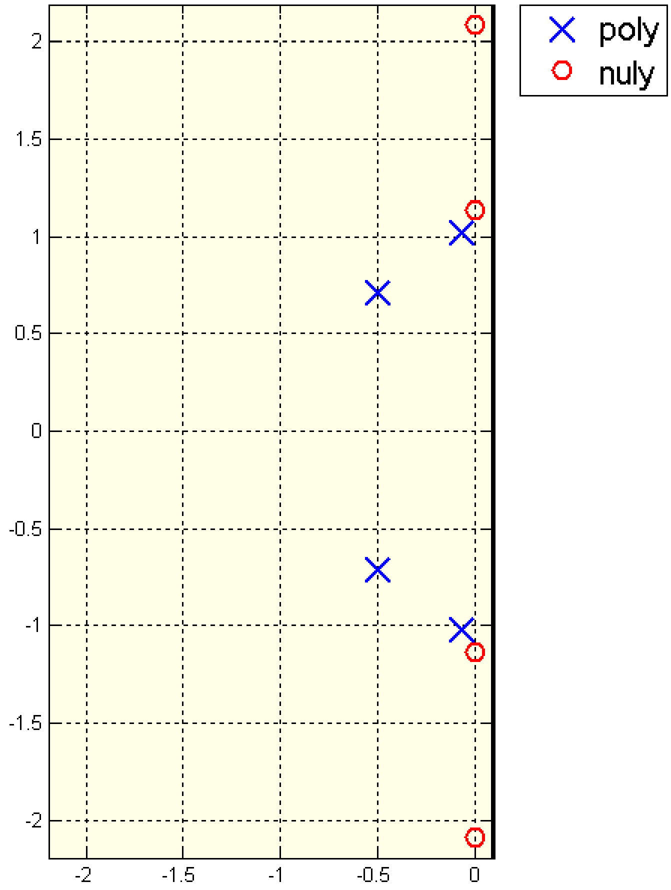
-0,062 241 131 555 15 - 1,020 633 337 711 78i

-0,497 541 472 669 61 + 0,708 544 851 892 26i

-0,497 541 472 669 61 - 0,708 544 851 892 26i

K =

0,131 761 362 290 97



Obr. 3.6 Ukázka rozložení nulových bodů a pólů Cauerova filtru 4. Řádu

V kapitole 3 bylo čerpáno z:[1],[2], [3]

4 Popis metod návrhu FIR filtru

4.1 Úvod do FIR filtrů

Číslicové filtry s konečnou impulsní charakteristikou (finite impulse response) mají konečný počet, tedy N (délka filtru) reálných koeficientů ve vektoru systémových konstant $\mathbf{h} = [h_n], \quad n \in \langle 0, N-1 \rangle$.

Diferenční rovnice mezi vstupem x_n a výstupem y_n je ve tvaru:

$$y_n = \sum_{k=0}^{N-1} x_{n-k} h_k \ . \tag{4.1}$$

K popisu v rovině z stačí pouze nulové body, které jsou přímo systémovými realizačními konstantami. *N-1* (řád filtru) -násobný pól v počátku zajišťuje fázový posun a absolutní stabilitu.

Přenosová funkce je ve tvaru:

$$H(z) = \sum_{n=0}^{N-1} h_n z^{-n} \ . \tag{4.2}$$

Frekvenční charakteristika filtru má tento výsledný tvar:

$$H(e^{j\omega T}) = \sum_{n=0}^{N-1} h_n e^{-j\omega n T} \tag{4.3}$$

Lze docílit lineární fáze v celém rozsahu kmitočtů, pokud bude impulsní charakteristika:

1. symetrická (sudá symetrie): $h_n = h_{(N-1-n)}$
   - je osově souměrná, osa symetrie prochází bodem $\frac{N-1}{2}$
2. asymetrická (antisymetrická, lichá symetrie): $h_n = -h_{(N-1-n)}$
   - je středově souměrná, prochází středem souměrnosti, který leží taktéž v bodě $\frac{N-1}{2}$.

Dále ve spolupráci s počtem vzorků filtru (sudá nebo lichá) se odvozují tyto 4 možné případy:

Typ 1 h[n] symetrická, N liché,

Typ 2 h[n] symetrická, N sudé,

Typ 3 h[n] asymetrická, N liché,

Typ 4 h[n] asymetrická, N sudé.

Hlavní výhody:

1. absolutní stabilita filtru typu FIR, neboť všechny póly systému vždy leží v počátku uvnitř jednotkové kružnice;
2. koeficienty přenosové funkce jsou zároveň hodnoty impulsní charakteristiky;
3. možnost získat lineární fázovou frekvenční charakteristiku v celém pásmu kmitočtů;
4. pokud je impulsní charakteristika symetrická nebo asymetrická, je skupinové zpoždění τ konstantní, takže všechny složky kmitočtů obdrží stejné zpoždění výstupu vůči vstupu po průchodu systémem;
5. velmi výhodné nasazení v aplikacích realizujících širokopásmové filtry, kde je snadný návrh;
6. dále jsou vhodné i pro adaptivní filtraci, kde je nezbytné často měnit koeficienty z důvodu změny tvaru či typu přenosové charakteristiky, a kde díky již zmíněné

stabilitě není nutné ji zpětně testovat a tím pádem lze zjednodušit celý algoritmus na pouze nezbytné kroky, což ušetří výpočetní sílu celého systému;

7. nižší citlivost na kvantování koeficientů a stavových veličin, které neovlivňuje stabilitu systému;
8. menší riziko zahlcení aritmetiky mikroprocesoru, který právě zpracovává signál;
9. relativně snadný návrh a realizace signálovým procesorem, či implementace v jazyce symbolických adres.

Nevýhody:

1. neexistuje vzájemná paralela v oblasti analogové techniky;
2. příliš vysoký řád *N-1* filtru, pokud realizujeme přenosovou funkci se strmými přechody mezi frekvenčními pásmy v tolerančním schématu (např. úzkopásmové filtry), paměťové nároky při výpočtu mohou být značné;
3. problematický a často komplikovaný návrh specifických útlumových požadavků v propustném i nepropustném pásmu tolerančního pole.

4.2 Metody návrhu:

4.2.1 Metoda váhové posloupnosti

Tato metoda vychází ze specifikace ideální frekvenční charakteristiky $H_d\left(e^{j\hat{\omega}}\right)$. Protože je tato charakteristika nekonečně dlouhá, zkrátíme ji na konečnou délku filtru N tak, aby dokázala s dostatečnou přesností uspokojivě nahradit frekvenční požadavky ideální charakteristiky. Tuto impulsní odezvu $h_d(n)$ násobíme vhodnou funkcí okna délky N. Pro danou specifikaci filtru vypočteme délku filtru N a volíme typ okna $w[n]$ s nejširším lalokem a s nejbližší vyšší hodnotou útlumu na základě specifikace.

$$H_d\left(e^{j\hat{\omega}}\right) = \begin{cases} 1 \cdot e^{-j\hat{\omega}}, & |\hat{\omega}| \le \hat{\omega}_p \\ 0 & \hat{\omega}_p \le |\hat{\omega}| \le \pi \end{cases} \tag{4.4}$$

$$h_w[n] = h_d[n]w[n] \tag{4.5}$$

Jméno okna	Maximální potlačení v nepropustném pásmu [dB]	Potlačení hlavního laloku [dB]	Normovaná šířka přechodového pásma Δω odvozená od délky filtru *N*
Obdélníkové (pravoúhlé)	21	13,3	*1,8π/N*
Barlettovo (trojúhelníkové)	25	26,5	*6,1π/N*
von Hannovo (Hanningovo)	44	31	*6,2π/N*
Hammingovo	53	41	*6,6π/N*
Blackmanovo	74	57	*11π/N*
Kaiserovo	➢ optimalizované okno, ➢ velká šířka hlavního laloku pro dané potlačení v nepropustném pásmu, ➢ při shodné délce filtru umožňuje zajistit různou šířku přechodového pásma, ➢ existují empirické vzorce pro odhad délky filtru viz [2].		

Tab. 4.1 Nejběžnější typy oken s parametry a vztahy pro výpočet přibližné délky filtru

Normovaná šířka přechodového pásma je učena:

$$\Delta\omega = \hat{\omega}_s - \hat{\omega}_p, \tag{4.6}$$

kde $\hat{\omega}_s, \hat{\omega}_p$ jsou mezní úhlové frekvence digitálního filtru nepropustného a propustného pásma v [rad].

Vlastnosti návrhu filtru pomocí okenních funkcí:

- relativně jednoduchá, přímočará a historicky nejstarší metoda,
- fázová kmitočtová charakteristika je vždy lineární,
- vhodné pro návrh širokopásmových filtrů,
- nepřesná aproximace hranic přechodových pásem,
- obtížný odhad vhodného řádu filtru a komplikovanější volba okenní funkce, pokud chceme dosáhnout obstojné aproximace frekvenční charakteristiky.

4.2.2. Metoda vzorkování kmitočtové charakteristiky

Základní myšlenka této metody je:

Využijeme ideální kmitočtovou charakteristiku filtru $H\left(e^{j\bar{\omega}}\right)$.

Vybereme řád filtru N a začneme vzorkovat $H\left(e^{j\bar{\omega}}\right)$v N ekvidistantních úhlových kmitočtech digitálního filtru v úplném kmitočtovém pásmu (0,2π). Tímto způsobem obdržíme diskrétní funkci *H(k)*. Výslednou impulsní charakteristiku je pak rovna:

$$h[n] = \frac{1}{N} \sum_{k=0}^{N-1} H(k) e^{j2\pi nk/N} \tag{4.7}$$

Pokud bude impulzní charakteristika vykazovat jeden ze čtyř druhů symetrie, jak bylo uvedeno v kap. 4.1, bude mít výsledný filtr lineární fázi.

Vlastnosti návrhu filtru pomocí vzorkování kmitočtové charakteristiky:

- přiměřenou volbou umístění vzorků lze upravit přenos kmitočtové charakteristiky v přechodovém pásmu,
- s výhodou lze použít pro úzkopásmové filtry, kdy počet vzorků v propustném pásmu je výrazně vyšší než počet vzorků v nepropustném pásmu.

4.3.3 Optimalizovaná metoda se stejnoměrným zvlněním

Metody návrhu pomocí oken a frekvenční vzorkování jsou docela jednoduché a lehce využitelné. Avšak skýtají určité nevýhody:

1. Při návrhu nelze přesně specifikovat kmitočty propustného a nepropustného pásma. Musíme se spokojit s hodnotami, které vrátí návrh.
2. Nelze specifikovat tolerance zvlnění odděleně
3. Není možné stejnoměrně rozložit tzv. chybu aproximace přes celé propustné a nepropustné pásmo. V místech nespojitosti požadovaného průběhu kmitočtové charakteristiky nastává největší zvlnění.

Maximální chybu aproximace lze minimalizovat postupem zvaným minimax metoda nebo také Čebyševova aproximace. Pokud filtry splňují soubor podmínek, pro který jsou optimální z hlediska minimální maximální chyby aproximace, tak mají vlastnost, ve které je chyba aproximace rozložena rovnoměrně v propustném i nepropustném pásmu.

Minimax problém

Parks a McClellan použitím trigonometrických vztahů převedli amplitudovou charakteristiku na jednotný tvar:

$$H_r(\widehat{\omega}) = R(\widehat{\omega})P(\widehat{\omega}) \tag{4.8}$$

$P(\widehat{\omega})$je aproximační funkce, která má obecný tvar:

$$P(\widehat{\omega}) = \sum_{m=0}^{L} \alpha(m) \cos \widehat{\omega} m \tag{4.9}$$

L značí řád aproximačního polynomu $\alpha(m)$ určuje parametry filtru. Existují čtyři případy symetrie, jež jsou uvedené v kap. (4.1)

Typ filtru FIR	R($\hat{\omega}$)	L	P($\hat{\omega}$)
1. typ	1	$\frac{N-1}{2}$	$\sum_{m=0}^{L} \tilde{a}(m)\cos\hat{\omega}m$
2. typ	$\cos\frac{\hat{\omega}}{2}$	$\frac{N}{2}-1$	$\sum_{m=0}^{L} \tilde{b}(m)\cos\hat{\omega}m$
3. typ	$\sin\hat{\omega}$	$\frac{N-3}{2}$	$\sum_{m=0}^{L} \tilde{c}(m)\cos\hat{\omega}m$
4. typ	$\sin\frac{\hat{\omega}}{2}$	$\frac{N}{2}-1$	$\sum_{m=0}^{L} \tilde{d}(m)\cos\hat{\omega}m$

Tab. 4.2 Čtyři případy symetrie pro hodnoty R($\hat{\omega}$),P($\hat{\omega}$) a L

Pokud chceme využít Čebyševovu aproximaci k minimalizaci chyby, musíme nadefinovat požadovanou amplitudovou charakteristiku $H_{dr}(\hat{\omega})$ a váhovací kmitočtovou funkci $W(\hat{\omega})$, která umožní nezávislé ovládání zvlnění δ_1 v propustném a δ_2 v nepropustném pásmu. Váženou chybovou funkci $E(\hat{\omega})$definujeme přes celé kmitočtové pásmo takto:

$$E(\hat{\omega}) = W(\hat{\omega})\left[H_{dr}(\hat{\omega}) - H_r(\hat{\omega})\right], \qquad \hat{\omega} \in S \equiv \left[0, \hat{\omega}_p\right] \cup \left[\hat{\omega}_s, \pi\right] \tag{4.10}$$

Váhovací funkci lze zadat ve tvaru:

$$W(\hat{\omega}) = \begin{cases} \dfrac{\delta_2}{\delta_1} & \text{v propustném pásmu} \\ 1 & \text{v nepropustném pásmu} \end{cases}$$

Tímto docílíme toho, že maximální chyba bude mít v obou pásmech velikost δ_2. Pokud dokážeme minimalizovat maximálně váženou chybu δ_2 tak splňujeme i specifikaci δ_1 v propustném pásmu.

Řešíme problém Čebyševovy aproximace a tedy hledáme koeficienty $\widetilde{a}$ (m), $\widetilde{b}$ (m), $\widetilde{c}$ (m), $\widetilde{d}$ (m) aproximační funkce P($\widehat{\omega}$), které minimalizují maximum chybové funkce $E(\widehat{\omega})$, neboli:

$$\min\{\max|E(\widehat{\omega})|\} \qquad \text{pro} \qquad \widehat{\omega} \in S \tag{4.11}$$

Velmi důležitý parametr optimalizace, který vykazuje $E(\widehat{\omega})$, je počet maxim a minim, tedy extrémů. Čebyševova aproximace pro návrh filtru FIR může být shrnuta do tzv. alternačního teorému.

Alternační teorém:

V případě, že je $P(\widehat{\omega})$lineární kombinací kosinových funkcí, platí tedy vztah (4.7), pak nutnou a postačující podmínkou, že $P(\widehat{\omega})$ je jedinou nejlepší váženou Čebyševovou aproximací zadané funkce $H_{dr}(\widehat{\omega})$ na intervalu <0,π) je, že vážená chybová funkce vykazovala alespoň (L+2) zákmitů na L+2 kmitočtech $\widehat{\omega}_i$ na S takových, že platí:

$$E(\widehat{\omega}_i) = -E(\widehat{\omega}_{i-1}) \qquad \text{pro} \qquad \widehat{\omega}_0 < \widehat{\omega}_1 < \ldots < \widehat{\omega}_{L+1} \in S \tag{4.12}$$

Většinou u těchto optimalizovaných filtrů narazíme na (L+2) extrémů zvlnění, ale u některé kombinace mohou mít až (L+3) extrémů.

Z alternační teorému je zřejmé, že lze nalézt pouze jedno řešení aproximace minimax, ale konkrétní řešení však nepřináší. Protože je jeho analytické řešení příliš složité, využívá se iterační postup, kde je nejznámější Remezův algoritmus. Využití aplikace Remezova výměnného algoritmu předpokládá znalost délky filtru a poměr δ_2 / δ_1. Pro odhad délky filtru *N* využijeme Kaiserův vzorec:

$$N = \frac{-20\log_{10}\sqrt{\delta_1\delta_2} - 13}{14.6\Delta f} + 1, \qquad \text{kde } \Delta f = \frac{\widehat{\omega}_s - \widehat{\omega}_p}{2\pi} \tag{4.13}$$

Iterační postup Remezova algoritmu:

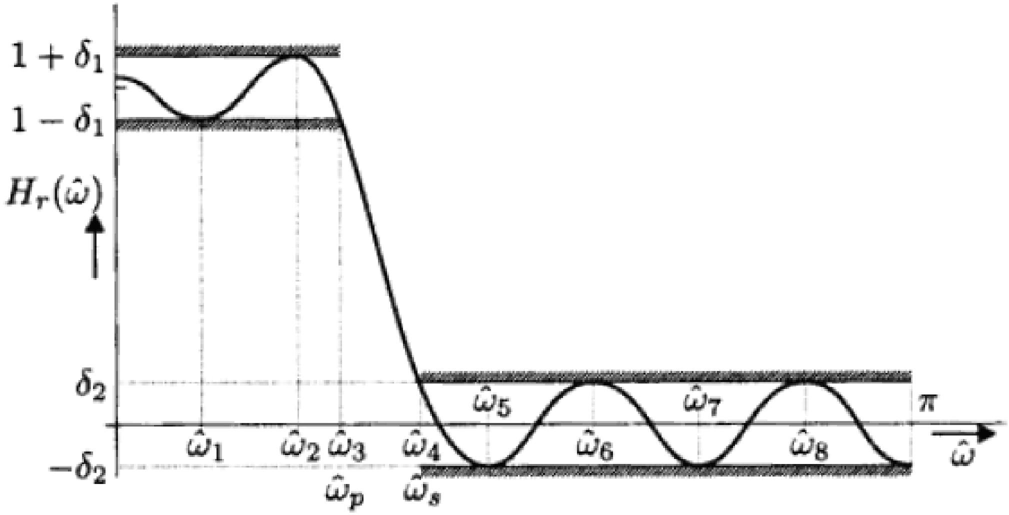
1) Provedeme prvotní odhad (L+1) kmitočtů extrémů $\widehat{\omega}_i$.
2) Provedeme odhad aproximační chyby pro zvolené kmitočty.
3) Z odhadů vypočteme koeficienty a sestavíme polynom $P(\widehat{\omega})$ *L*-tého řádu v kmitočtech $\widehat{\omega}_i$.
4) Vypočítáme lokální maxima *E* a v těchto extrémech určíme nové $\widehat{\omega}_i$.
5) Vypočteme polynom $P(\widehat{\omega})$ *L*-tého řádu v nových kmitočtech $\widehat{\omega}_i$.
6) Dokud nenalezneme množinu optimálních frekvencí $\widehat{\omega}_i$ a hodnotu chyby δ proceduru opakujeme.
7) Zvětšíme řád *N*, pokud je $\delta < \delta_2$. V opačném případě řád *N* zmenšíme a znovu opakujeme Remezův algoritmus a určíme novou hodnotu δ .Postup opakujeme do té doby dokud $\delta_2 \geq \delta$.

Vlastnosti metody rovnoměrně zvlněných aproximací

1) \lze dosáhnout nejmenší šířky přechodového pásma a nejnižší řád filtru můžeme přesně dodržet kmitočtové hranice propustného a nepropustného pásma.
2) Neexistuje přesný analytický vzorec pro určení dostatečného řádu filtru.
3) Aby byla zajištěna konvergence algoritmu, je nutné zajistit dostatečnou přesnost výpočtů pro iterační postupy.

Obr. (4.1) znázorňuje amplitudovou charakteristiku H_r dolní propusti filtru FIR typu 1a zarovň jsou zde dle alternačního teorému vyznačeny extrémy.

Uvedená ilustrace je převzata z [3].



Obr. 4.1 Typický příklad aproximace podle alternačního teorému

V kapitole 4 bylo čerpáno z:[1],[3],[5]

5 Konečná délka slova

5.1 Úvod

Tutéž frekvenční charakteristiku lze realizovat několika rozdílnými strukturami implementace filtru. V případě výpočetního systému s konečnou přesností se tyto struktury chovají odlišně. Ve většině případů mají hodnoty signálů v procesu číslicového zpracování (např. lineární kmitočtová filtrace) vždy konečnou délku slova a do paměťových registrů se ukládají často v lineárním tvaru. Důsledkem toho se lineární charakter původně navrženého číslicového filtru může měnit na nelineární a odezva takového filtru se může odlišovat od požadované.

5.2 Metody zobrazení čísel

5.2.1 Pevná řádová čárka

Mezi nejvíce využívané polyadické číselné soustavy patří dvojková. Tato soustava má základ Z = 2 a každé číslo je tvořeno posloupností číslic, které mohou nabývat hodnot od 0 až po $Z - 1$, tedy pouze 0, 1. Každá číslice je charakteristická svou váhou 2^i a lze tedy jakékoliv reálné číslo A vyjádřit jako součet mocnin o základu 2.

$$A = \sum_{i=-\infty}^{k} a_i 2^i \quad , \quad a_i = 0 \text{ nebo } 1. \tag{5.1}$$

Nicméně kvůli omezené délce registru v číslicových systémech máme omezený počet zobrazitelných čísel a proto i omezenou přesnost zobrazení. Pro délku registru B lze vyjádřit předchozí vztah ve tvaru:

$$A = \sum_{i=-b}^{k} a_i 2^i = a_k 2^k + a_{k-1} 2^{k-1} + \ldots + a_1 2^1 + a_0 + a_{-1} 2^{-1} + \ldots + a_{-b} 2^{-b} \tag{5.2}$$

kde k označuje nejvyšší váhu celočíselné části čísla A a b řád nejméně významného bitu, kterým je určen kvantizační krok q, jenž popisuje rozlišení dvou sousedních čísel.,

$$q = 2^{-b}\,. \tag{5.3}$$

V registru délky B lze zobrazit celočíselnou, i zlomkovou část čísla. Délku registru B lze zapsat vztahem:

$$B = k + b + 1\,. \tag{5.4}$$

Řádová čárka čísla A je umístěna za členem a_0 a rozděluje číslo na celočíselnou část (*integer*) A_i a zlomkovou (*fractional*) A_f.

$$A = A_i + A_f = \sum_{i=0}^{k} a_i 2^i + \sum_{i=1}^{b} a_{-i} 2^{-i} \tag{5.5}$$

Poloha řádové čárky v registru je pouze imaginární. Je ale nutné, aby byla dodržena a respektována pro všechny registry a všechny prováděné výpočty. Nejčastěji se setkáváme s řádovou čárkou umístěnou buď za posledním bitem zprava, což odpovídá celočíselné reprezentaci (integer) nebo za prvním bitem zleva, a potom se jedná o zlomkovou reprezentaci čísel (fraction).

Velikost zobrazovaných čísel je závislá nejenom na délce registru ale i na umístění řádové čárky. Nevýhodou z hlediska reprezentace čísel v pevné řádové čárce je skutečnost, že počet zobrazitelných čísel je výrazně nižší, než je tomu v případě pohyblivé řádové čárky. Pro popis reprezentací se používá parametr tzv. dynamický rozsah zobrazení. Tento parametr se udává v decibelech a je definován jako poměr největšího a nejmenšího zobrazitelného čísla. Dynamický rozsah pro pevnou řádovou čárku s délkou registru B určuje vztah:

$$D_{fix} = 20\log\frac{A_{\max}}{A_{\min}} - 20\log\frac{2^{B}-1}{2^{0}} \approx 6.02\cdot B \quad [\mathrm{dB}]. \tag{5.6}$$

Dynamický rozsah lze zjednodušit na poučku, kdy na jeden bit připadá zhruba 6 dB. Mezi další nevýhody pevné řádové čárky patří malý dynamický rozsah zobrazovaného čísla a chyby kvantování závislé na velikosti zobrazovaného čísla. Absolutní hodnota kvantizačního kroku je tedy neměnná a nezávislá na pevné délce registru. Naproti tomu relativní velikost kvantizační chyby vykazuje závislost nepřímo úměrnou hodnotě zobrazovaného čísla.

5.2.2 Pohyblivá řádová čárka

V registru s danou délkou lze pomocí pohyblivé někdy nazývané plovoucí čárky zvětšit rozsah zobrazitelného čísla. Je to možné, protože v tomto registru není přiřazena pevná váha 2^{i}, ale váha proměnná. Číslo A je potom možné stanovit pomocí dvojice čísel (M, E):

$$A = M\cdot 2^{E}. \tag{5.7}$$

M označujeme jako mantisu a E jako exponent. Exponent je kladné, nebo záporné číslo udávající místo řádové čárky v mantise. Absolutní hodnota mantisy je většinou větší než jedna, protože se jedná o zlomkové binární číslo, které má řádovou čárku za prvním bitem zleva. Registr dokáže zobrazit jak mantisu tak exponent. Každá jeho část má vyhrazen určitý počet bitů. Dle normy formátu IEEE 754 má mantisa 24 bitů a exponent je osmibitové číslo. Tento formát označujeme *single* nebo *float*.

Zobrazení téhož čísla s plovoucí řádovou čárkou lze provést několika způsoby, je totiž nejednoznačné. Tuto nevýhodu eliminuje proces normování mantisy. Normovaná mantisa nabývá výhradně hodnot v intervalu $\langle 0{,}5;1)$ s tím, že první dva bity zleva jsou invertované.

Mezi výhody reprezentace pomocí řádové čárky patří velký rozsah zobrazitelných čísel. Dynamický rozsah pro pohyblivou čárku D_{float} lze popsat vztahem:

$$D_{float} = 6{,}02 \cdot \left(2^{B_E} - 1\right) \tag{5.8}$$

B_E zde značí počet bitů v exponentu.

Mezi nevýhody plovoucí řádové čárky patří složité a časově náročné provádění početních operací. Pokud násobíme dvě čísla tak nejprve násobíme jejich mantisy a sčítáme jejich exponenty. Poté normujeme výsledné číslo v souladu s podmínkou existence hodnot pouze v intervalu $\langle 0{,}5;1)$. Avšak pokud dvě čísla sčítáme je nutné nejprve exponenty obou čísel upravit na stejnou hodnotu. Teprve potom je možné mantisy obou čísel sečíst a normovat.

5.2.3 Záporná čísla

Záporná čísla lze zpravidla popsat těmito metodami:

1. zobrazení se znaménkovým bitem
2. zobrazení s dvojkovým doplňkem (doplňkový kód)
3. zobrazení s jedničkovým doplňkem (inverzní kód)

Pokud chceme do registru uložit kladná i záporná čísla, musíme jej rozdělit na dva zhruba stejně velké díly. Uvedené reprezentace lze aplikovat stejnou měrou nejen na pevnou řádovou čárku, ale též i na mantisu v pohyblivé čárce.

Zobrazení se znaménkovým bitem

Jakékoliv číslo lze popsat jeho absolutní hodnotou a znaménkem, pro které je v tomto případě vyčleněn nejvyšší bit, tzv. znaménkový bit X_S Pro kladná čísla je nulový, pro záporná je roven jedné. Výhodou je jednoduchost zobrazení navzdory složitým početním operacím, obzvláště sčítání.

Dvojkový doplněk

Dvojkový doplněk je nejvíce využívaný způsob reprezentace záporného čísla využívající rozdílu kladného čísla od čísla 2 neboli $(1\ 0)_2$.v binární soustavě. Toho docílíme, pokud invertujeme všechny bity a nejméně významný bit zvýšíme o log1.

Dvojkový doplněk lze k danému binárnímu číslu vyjádřit pomocí vzorce:

$$X = X_M \left(-X_S + \sum_{i=1}^{b} X_i 2^{-i} \right). \qquad (5.9)$$

X_M slouží jako měřítko vyjádřené jakoukoliv celočíselnou konstantou.

Jedničkový doplněk

Jedničkový doplněk získáme prostou inverzí všech číslic kladného čísla. Význačná je zde souměrnost čísel podle nuly a nejvyšší váhu má opět znaménkový bit. Tab. 5.1 je převzata z [1].

Dvojkové číslo	Znaménko + mantisa	Inverzní kód	Doplňkový kód
0 1 1 1	7/8	7/8	7/8
0 1 1 0	6/8	6/8	6/8
0 1 0 1	6/8	5/8	5/8
0 1 0 0	4/8	4/8	4/8

0 0 1 1	3/8	3/8	3/8
0 0 1 0	2/8	2/8	2/8
0 0 0 1	1/8	1/8	1/8
0 0 0 0	+ 0	+ 0	+ 0
1 0 0 0	- 0	- 7/8	- 1
1 0 0 1	- 1/8	- 6/8	- 7/8
1 0 1 0	- 2/8	- 5/8	- 6/8
1 0 1 1	- 3/8	- 4/8	- 5/8
1 1 0 0	- 4/8	- 3/8	- 4/8
1 1 0 1	- 5/8	- 2/8	- 3/8
1 1 1 0	- 6/8	- 1/8	- 2/8
1 1 1 1	- 7/8	- 0	- 1/8

Tab. 5.1 Porovnání třech typů zobrazení v pevné čárce

Z uvedených zákonitostí vyplývá:

- kladná čísla jsou ve všech kódech vyjádřena stejně;
- dvojkový doplněk má největší rozsah, protože zbývající dvě reprezentace zobrazují nulu dvojím způsobem „-0“ a „+0“;
- jako jeho další výhodu můžeme uvést důležitou vlastnost, že dojde-li při sčítání nebo odčítání k překročení rozsahu zobrazení, pak nedojde k chybě, ale jen pokud se při následující početní operaci výsledek korektně navrátí zpět do původního rozsahu

5.3 Kvantovaní v číslicových systémech

Přesnost zobrazovaných čísel je závislá na velikosti registru B s rozlišením $X_M 2^{-b}$ a není tedy nekonečná. Toto omezení přesnosti označujeme jako kvantování. Kvantování lze provést dvěma způsoby:

- zaokrouhlením
- odříznutím

Tyto operace mají nelineární charakter a během nich do hodnot jednotlivých kvantovaných signálů přidávají dříve nepřítomnou kvantizační chybu. Kvantizační chyba je rozdíl mezi již kvantovaným signálem X_Q a signálem X v plné přesnosti.

$$e = X_Q - X \tag{5.10}$$

Kvantizační chyba způsobená zaokrouhlením e_R má toto rozmezí hodnot:

$$-\frac{q}{2} \leq e_R \leq \frac{q}{2} \tag{5.11}$$

Kvantizační chyba způsobená odříznutím e_T má toto rozmezí hodnot:

$$-q \leq e_T \leq 0 \tag{5.12}$$

5.3.1 Přetečení

Pokud překročíme rozsah zobrazovaného čísla v číslicovém signálu dojde k tzv. přetečení. Snažíme se tedy do systému zanést metodu k jeho ošetření. Nejčastější metodou ošetření je tzv. saturace. Při překročení rozsahu dostaneme na výstupu maximální hodnotu. Další metodou, která bere v úvahu riziko vzniku limitního cyklu je metoda nulování. V tomot případě dostaneme na výstupu hodnotu nula.

5.3.2 Kvantování amplitudy vstupního signálu

Než vznikne číslicový signál je třeba původní analogový signál navzorkovat. Získáme diskrétní vzorky s nekonečnou přesností. V tomto stavu nelze vzorkovaný signál uchovat, a proto je nutné amplitudu těchto vzorku kvantovat, tak jak to provádí *A/D* převodník. Na výstupu získáme kvantovaný signál a zároveň vznikne kvantizační chyba. Hodnoty vstupních vzorku musí ležet uvnitř rozsahu čísel. Z toho důvodu musí být vstupní signál normován. Takto omezujeme chyby přetečení.

5.3.3 Kvantování koeficientů filtru

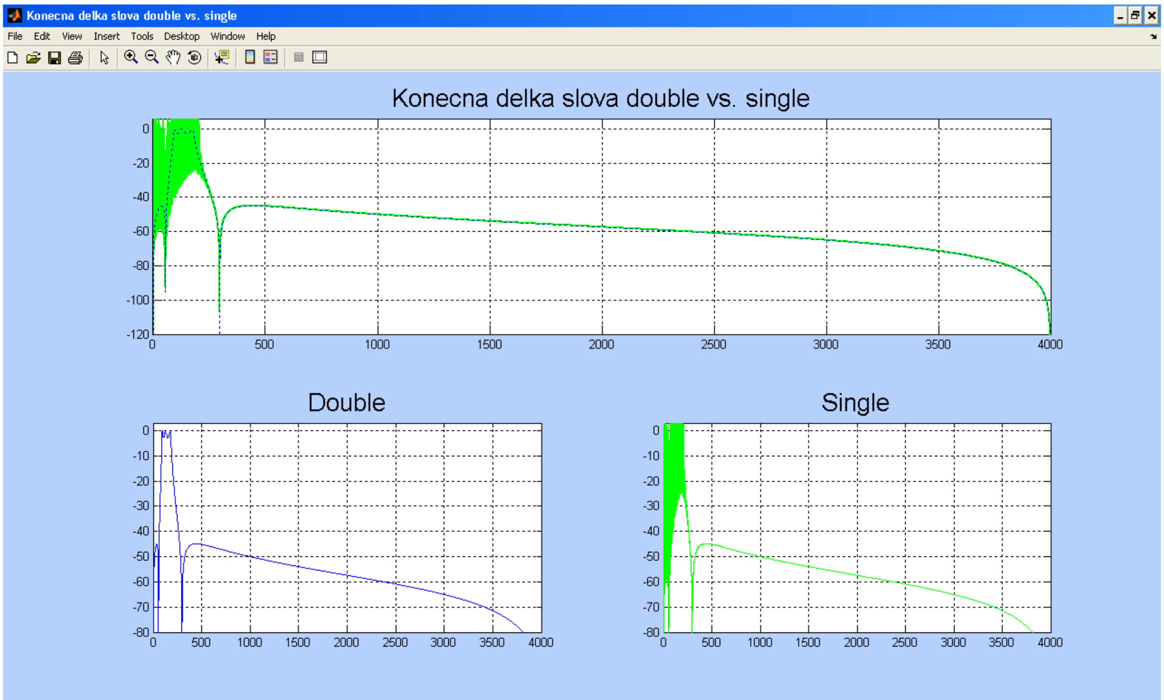
Koeficienty číslicových filtrů zpravidla navrhujeme s vysokou přesností. Realizace filtru vyžaduje tyto přesné hodnoty kvantovat. Kvantování má za následek změnu polohy nul a pólů v *z*-rovině. Tím pádem může dojít ke změně frekvenční charakteristiky navrženého filtru oproti charakteristice filtru reálného. Vliv kvantování koeficientů může způsobit posun některého z pólů do blízkosti jednotkové kružnice. Filtr se může stát nestabilním. Každá struktura filtru reaguje na vlivy kvantování koeficientů odlišně. Citlivost na kvantování koeficientů lze snížit zmenšením počtu koeficientů.

U přímé struktury filtru ovlivňuje hodnota všech koeficientů jmenovatele každý pól a hodnota všech koeficientů čitatele všechny nulové body. Již malé změny čitatele a jmenovatele dokážou způsobit velké změn poloh pólů nul. Největší riziko hrozí pokud nuly či pólu leží v bezprostřední blízkosti jako úzkopásmových filtrů.

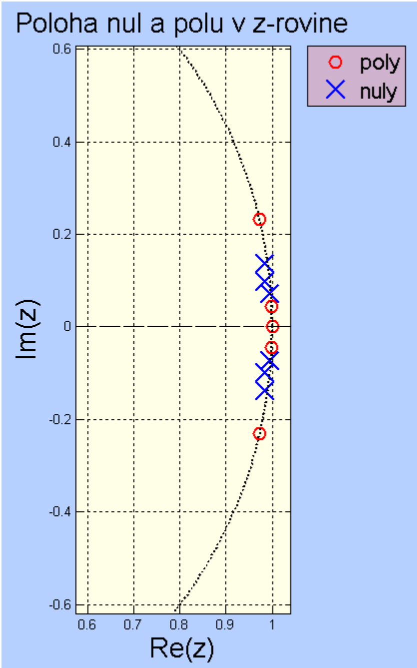
Paralelní struktura má nižší počet koeficientů v každé větvi. Nuly jsou ovlivňovány všemi koeficienty. Kvantování koeficientů jen nepatrně ovlivní tvar propustného pásma, ale stane se velmi citlivým v nepropustném pásmu, které je určeno polohou nulových bodů.

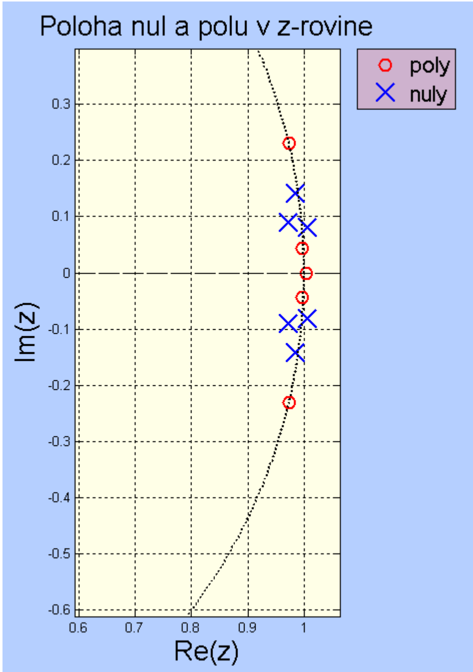
Kaskádní struktury vytvářejí sekce prvního druhého řádu, proto filtry s touto strukturou mají malou citlivost, jak v propustném tak v nepropustném pásmu. Z toho důvodu je tato realizace nejčastější.

Speciální struktury (příčkové, stavové a vlnové filtry) mají za úkol navrhovat číslicové filtry s nejnižším efektem vlivu konečné délky slova. Vliv kvantování a náchylnost k limitním cyklům je snižována na minimum.



Obr. 5.1 Ukázka vlivu konenčné délky slova formátu DOUBLE vs. SINGLE





Obr. 5.2 Ukázka nulových bodů a pólů v přesnosti DOUBLE vs. SINGLE

Výpis koeficientů:

DOUBLE reprezentace:

čitatel =

0,001 336 566 482 28

-0,005 271 159 227 89

0,006 532 756 585 56

0,000 000 000 000 03

-0,006 532 756 585 59

0,005 271 159 227 91

-0,001 336 566 482 28

jmenovatel =

1,000 000 000 000 00

-5,923 470 174 284 02

14,653 741 942 095 27

-19,378 610 574 768 27

14,448 448 440 966 18

-5,758 674 647 264 17

0,958 566 048 648 70

SINGLE reprezentace:

čitatel =

0,001 336 6

-0.005 271 2

0,006 532 8

0,000 000 0

-0,006 532 8

0,005 271 2

-0,001 336 6

jmenovatel =

1,000 000 0

-5,923 470 0

14,653 741 8

-19,378 610 6

14,448 448 2

-5,758 674 6

0,958 566 1

Limitní cykly v rekurzivních číslicových filtrech

Pokud budeme číslicový filtr považovat za lineární systém, pak bude jeho odezva totožná s požadovanou nezávisle na operacích. V případě stabilního systému bude jeho impulzní odezva postupně utlumena. V běžném reálném systému funguje konečná délka slova, proto může dojít k překročení této délky. Lineární systém se stává nelineárním a jeho odezva začne vykazovat oscilaci. Vznik limitních cyklů mají na svědomí zpravidla tyto příčiny vzniku:

- přetečení při součtech
- kvantování výsledků součinů

Limitní cyklus způsobí na výstupu filtru signál aniž by byl signálem vybuzen, označen též jako tzv. limitní cyklus při nulovém vstupu. Limitní cyklus vzniklý při kvantování součinů způsobí na výstupu oscilace, které vznikají vlivem omezení výsledku násobení ve zpětných vazbách. Chyby ze zaokrouhlování nebo zkrácení výsledku se kumulují, až přerostou v limitní cyklus. Filtry bez zpětnovazební smyčky tímto problémem netrpí.

Limitní cyklus přetečení může nastat pouze v případě součtu. V rekurzivním systému vlivem určité kombinace hodnot koeficientů a vstupních podmínek může dojít k oscilaci na výstupu filtru i při nepřítomnosti vstupního signálu. Vhodná aritmetika s využitím saturace dokáže mnohdy tyto problémy odstranit.

V kapitole 5 bylo čerpáno z:[1],[3]

6 Grafické uživatelské prostředí a rozhraní

Obecný postup návrhu GUI

Nejjednodušší způsob jak vytvořit GUI je postupně přidávat komponenty a rozšiřovat či je modifikovat dle požadavků vyvstávajících z aktuálního problému. Na druhou stranu pro komplikovanější systémy je tato metoda naprosto nevhodná. A proto je pro složitější a rozsáhlejší projekty výhodné postupovat dle tohoto doporučení:

1. Zásadní je hlubší rozvaha jednotlivých prvků před vlastní tvorbou GUI, je třeba udělat hrubý náčrt, jaké máme představy o rozmístění prvků na ploše, jaké formy výstupu vyžadujeme a jaké funkce jednotlivých částí musíme vyřešit.
2. Umístíme, naklonujeme a nastavíme veškeré navržené objekty
3. Spustíme vygenerování GUI a zkontrolujeme, zda nám tento výstup a jeho vzhled vyhovuje
4. Přes „Property Inspector“ stanovíme počáteční podmínky, hodnoty a stav proměnných, pojmenujeme komponenty, vše závisí na uživateli.
5. Na závěr vložíme zpětné vazby „callbacks“ a oživíme a ověříme GUI.

Metoda návrhu GUI pomocí GUIDE

Prostředí MATLAB nabízí pro vytváření grafických uživatelských rozhraní (GUI) sadu nástrojů včetně interaktivního vývojového prostředí GUIDE. Zde můžeme pouhým přetažením myši několika objektů na plochu sestavit požadované prostředí v tzv. „Layout Editoru“. Vzhled jednotlivých komponent, rozměry, barvy a další parametry v tomto editoru také snadno upravíme. GUIDE je v podstatě automatický generátor GUI, který automaticky vygeneruje M-file a soubor **figure.**

Metoda „Switched Board Programming“ (SBP)

Tato metoda staví na příkazu **switch**, který umožňuje směrování reakcí na základě zpětné vazby. GUI začíná povinným příkazem „**function jmeno(vstpar)“** se jménem naší funkce a vstupním parametrem. Za příkazem „**if nargin==0**“bude kód všech grafických objektů aplikace. Pokud bude aplikace spuštěna se vstupním parametrem, nebude splněna podmínka a začne se vykonávat kód uvedený za částí **else**. Následuje stěžejní příkaz „**switch(vstpar)**“, který má určitý počet příkazů **case**. Program provede skok na takový **case**, který má u sebe uveden parametr odpovídající parametru ve zpětné vazbě. Klíčové u této metody je vlastnost volání funkce sebe sama s různými parametry.

Seznam kroků návrhu GUI metodou SBP je podstatně kratší a má totožný pouze první bod s metodou předchozí:

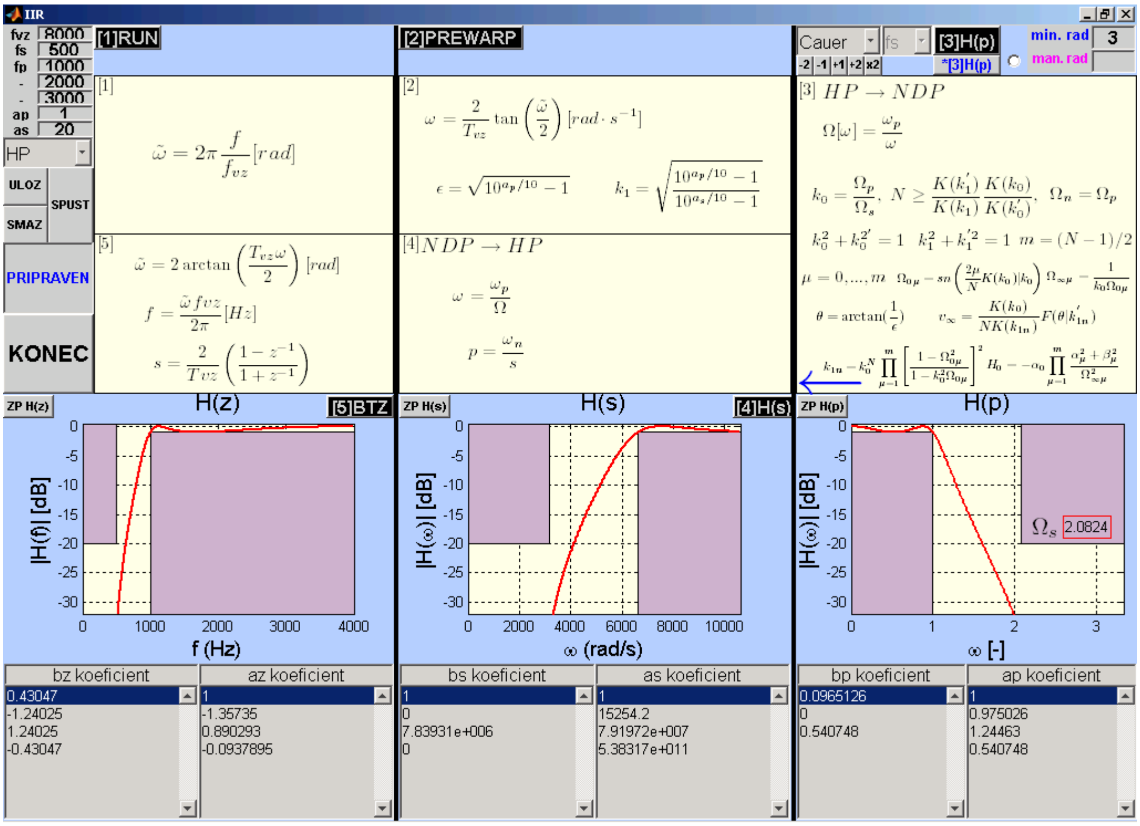
1. Rozvaha jednotlivých prvků před vlastní tvorbou GUI,
2. Veškeré grafické objekty a „callbacks“ ručně naprogramujeme do patřičných částí programu,
3. Ověříme požadované funkcionality a případně provedeme korekce

6.1 Tvorba GUI IIR

Jako první krok jsem si vypracoval odpovědi na následující otázky:

1. Jakou použít metodu zpracování programu,
2. jaký vzhled GUI a ovládání zvolit,
3. kterými komponenty zajistit zmíněné ovládání,
4. jaké budou odezvy na zásah uživatele,
5. jakými grafickými a datovými výstupy realizovat metodiku návrhu filtru.

Pro vlastní GUI jsem se po předchozích zkušenostech rozhodl využít metodu „**Switched Board Programming**“.



Obr. 6.1 Celkový pohled na GUI IIR

6.1.1 Rozvržení ovládacích prvků

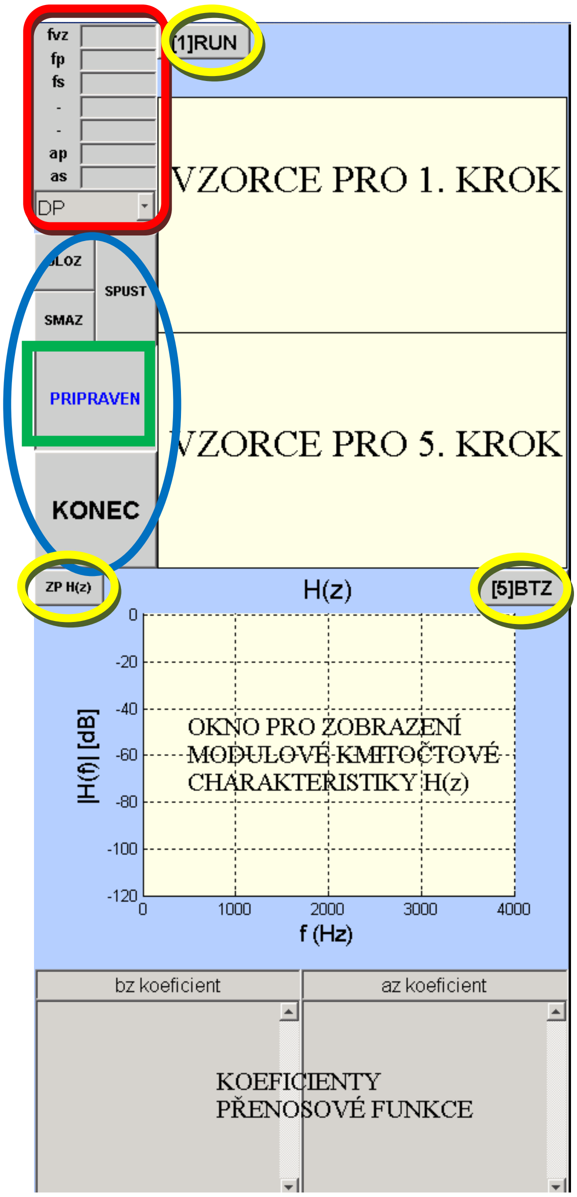
Program GUI IIR využívá:

- hlavní a 4 dočasná okna **figure**;
- **devět editačních polí** pro specifikaci tolerančního schématu a zobrazení indikace činnosti GUI, minimálního doporučeného řádu filtru nebo pro jeho manuální zadání;
- **18 tlačítek** pro uložení koeficientů přenosové funkce *H(z)* do souboru na disk, pro spuštění všech kroků návrhu postupně po krátkých časových intervalech, tlačítko pro kompletní vymazání vstupních parametrů, všech grafů, vzorců a koeficientů, pro zobrazení polohy nul a pólů přenosové funkce *H(p)*, *H(s)* a *H(z)*, tlačítka pro jednotlivé kroky návrhu, shluk tlačítek k snížení/zvýšení řádu filtru, tlačítko pro zobrazení rozsáhlejších vzorců a pro kompletní ukončení práce v GUI);
- **trojici výběrových menu** (umožňují volbu typu, aproximující funkce filtru a dodatečné dopřesnění specifikace u Butterwortha);
- **zaškrtávací políčko** (pro volbu mezi automatickým nebo manuálním řádem filtru);
- **osm polí axes** (pro zobrazení grafů tolerančního schématu, modulové frekvenční charakteristiky a použitých výpočetních vztahů);
- **šest rolovatelných polí „listboxů“** (pro zobrazení koeficientů přenosových funkcí *H(p)*, *H(s)* a *H(z)*).

Hlavní okno na obr. 6.1 lze z hlediska návrhu rozdělit na celkem tři samostatné celky pokrývající jeho jednotlivé fáze:

1. celek tvoří objekty zajišťující základní běh programu a také ty, které se přímo týkají návrhu v z-rovině. Obsahuje tyto následující komponenty, které spolu tvoří funkční celek nebo připravují hodnoty pro následující celek:

- soubor tlačítek pro běh programu označených modrou barvou
- indikátor činnosti/nečinnosti ve zeleném rámečku
- editační políčka a výběrové menu v sekci vstupních parametrů vyznačených červenou barvou
- tlačítka **[1]RUN**, **[5]BTZ**, **ZPH(z)** barvou žlutou
- 3x pole axes
- 2x rolovatelné pole vyčleněné pro koeficienty



Obr. 6.2 Vzhled 1.celku

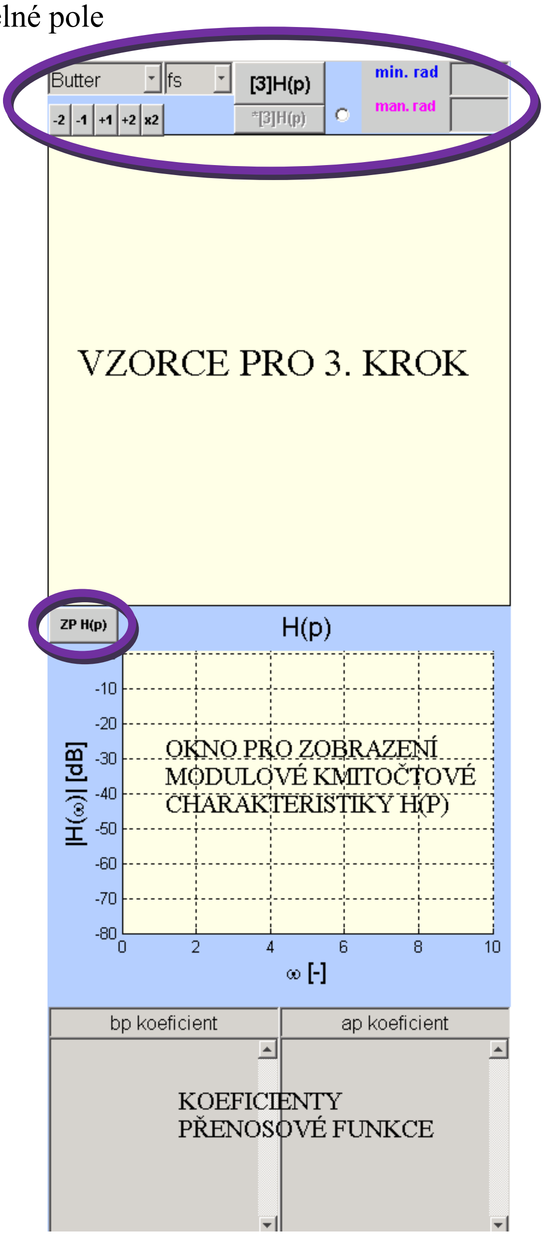
2. celek tvoří objekty zajišťující návrh analogového prototypu v s-rovině a skládá se z těchto následujících komponent:

- tlačítka **[2]PREWARP**, **[4]H(s)**, **ZPH(s)** znázorněna hnědou barvou
- 3x pole axes
- 2x rolovatelné pole vyčleněné pro koeficienty

Obr. 6.3 Vzhled 2.celku

3. celek tvoří objekty zpravující převod na NDP návrh v s-rovině a skládá se z těchto následujících komponent:

- 2x výběrové menu
- tlačítka **[3] H(p)**, ***[3] H(p)**, **ZPH(p)** a **{-2, -1, +1, +2, x2}**
- zaškrtávací a editační políčka pro ovládání řádu filtru
- výše zmíněné objekty jsou vyznačeny fialovou barvou
- 2x pole axes
- 2x rolovatelné pole



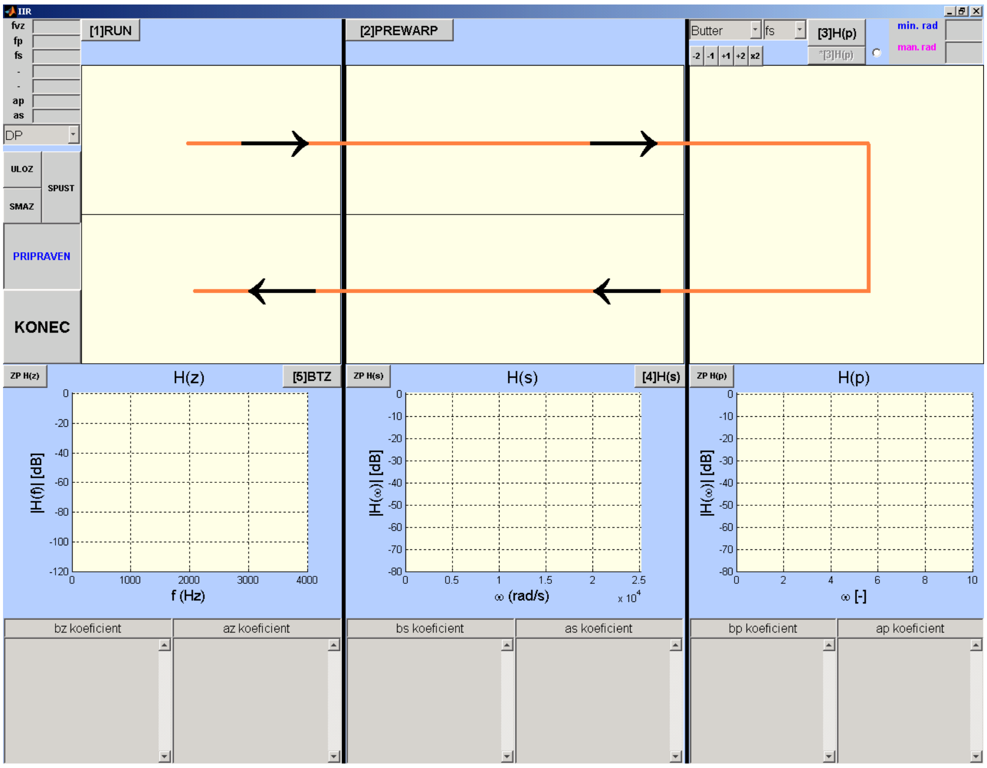
Obr. 6.4 Vzhled 3.celku

6.1.2 Popis funkce GUI

Grafické uživatelské rozhraní můžeme spustit přímo z příkazového řádku zadáním „IIRF“ nebo dvojklikem myší do aktuální složky v prostředí MATLAB na soubor IIRF.m. Otevře se hlavní okno **figure** a v levém horním rohu můžeme postupně v jednotkách hertz zadat:

- vzorkovací frekvenci f_{vz},
- frekvence pro propustné a nepropustné pásmo, nastavujeme od shora dolů, od nejnižší po nejvyšší hodnotu v souladu se značením a v závislosti na typu filtru (dolní, horní, pásmová propust nebo pásmová zádrž), který je vhodné navolit jako úplně první.
- dále v decibelech přesně stanovíme požadované útlumy vztažené k těmto již nadefinovaným pásmům.

Veškerý návrh filtru pokrývá pět funkčních tlačítek, která jsou úzce spjata s jednotlivými kroky, které se provádějí ve směru šipky od levého okraje okna k pravému a zpět viz obr. 6.5.



Obr. 6.5 Vzhled GUI včetně návrhové šipky

Obr. 6.6 Indikace stisku tlačítka [1]RUN

Po úvodním nadefinování tolerančního schématu přistoupíme ke stlačení tlačítka [**1**]**RUN**, které se přebarví jako na obr. 6.6. Vykoná se 1. krok návrhu, který spočívá ve vykreslení tolerančního schématu výsledného číslicového filtru do grafu „H(z)“ pro tuto chvíli ještě bez křivky aproximace. Současně se vyobrazí i vzorec pro stanovení normované úhlové frekvence [rad], jenž využívá následující krok. Frekvenční osa f [Hz] je ukončena na tzv. Nyquistově kmitočtu $f_N = f_{vz}/2$, osa y představuje modulovou frekvenční charakteristiku $|H(f)|$ [dB].

[2]PREWARP pracuje s grafem týkající se již analogové oblasti. Zde je nekonečná osa x [rad/s] ve skutečnosti zkrácena na konečnou množinu hodnot. 2. krok má obdobný úkol jako „1.“, ale s tím rozdílem, že vše náleží do roviny s. Zásadní je zde předzkreslení kmitočtové osy pro všechny kmitočty vytyčující toleranční schéma. Tato úprava je nutná pro závěrečný 5. Krok. Současně se se vzorcem náležícím 2. tlačítku objeví i vzorce pro tzv. sekundární parametry tolerančního schématu, které jsou nepostradatelné pro výpočet aproximační funkce.

Obr. 6.7 Příklad volby manuálního řádu

3. krok je klíčový. Nejdříve tedy zvolíme požadovanou aproximaci (Butterworthova, Čebyševova 1. druhu, Čebyševova inverzní 2. druhu a Cauerova eliptická), která předurčí vyobrazení příslušných vzorců. V případě Butterwortha máme na výběr zda splnit specifikaci na frekvenci Ω_p nebo Ω_s, v GUI značeno f_p, f_s ve výběrovém menu. Po stisku tlačítka **[3]H(p)** se nejprve provede tzv. dopředná frekvenční transformace a GUI nám doporučí minimální požadovaný řád aproximace.

V pravém horním rohu obdržíme takový nejnižší řád filtru tak, aby bylo vyhověno požadavkům vyplývajících z tolerančního schématu. Na základě zadaného typu a aproximace filtru se sestaví NDP a získáme koeficienty přenosové funkce normovaného analogového prototypu *H(p)*. O políčko níže lze dle obr. 6.7 vyzkoušet i jiný, než-li navržený minimální řád. Rolovatelná pole se naplní koeficienty a vykreslí se křivka přenosové funkce do „grafu H(p)“ a rovněž se do útlumového obdélníku nepropustného pásma vyobrazí číselný údaj o Ω_s. Osa *x* představuje normovaný úhlový kmitočet, *y*-ová osa je totožná jako u „grafu H(s)“.

Zde vzorce popisují:

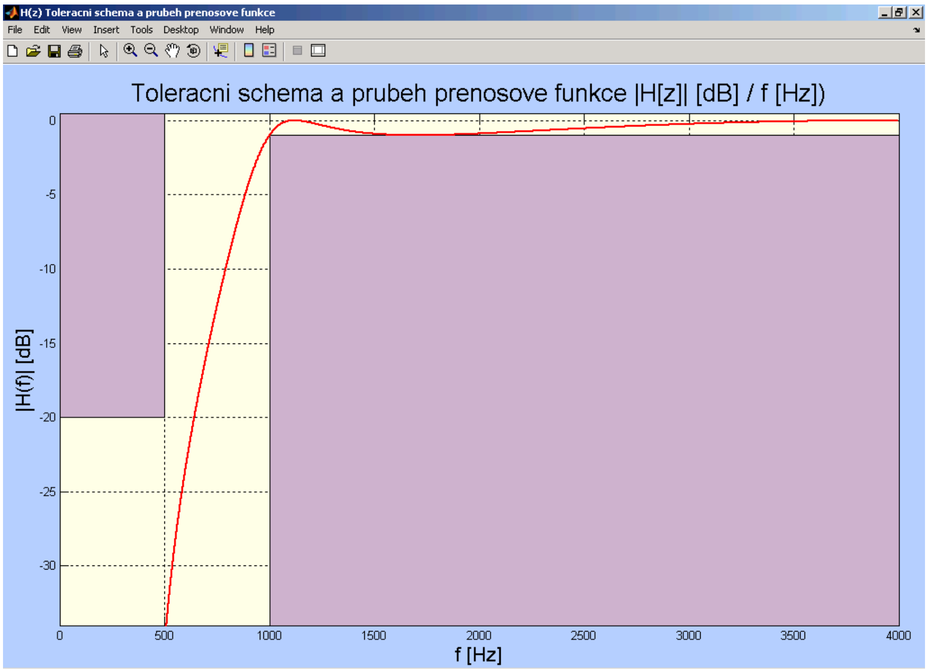
- převod na NDP,
- odhad řádu filtru,
- příprava charakteristické frekvence pro následné odnormování,
- výpočet přenosové funkce H(p).

V některých aproximacích (všechny kromě Butterworthovy) se uplatňují rozsáhlejší vzorce, a proto vyžadují více plochy k zobrazení. K tomuto účelu jsem vytvořil tlačítko ***[3]H(p)**, které dočasně využije zobrazovací plochu příslušející 4. kroku.

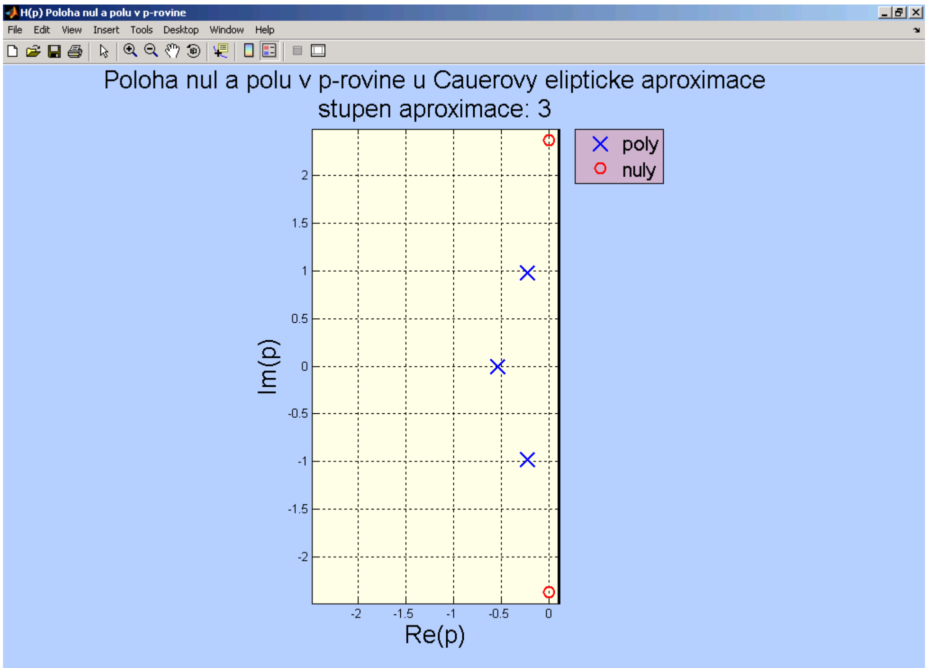
Na základě typu navrhovaného filtru se odvíjí tzv. zpětná frekvenční transformace neboli 4. krok, tlačítko **[4]H(s)**. Zde je spuštěna procedura odnormování na požadovaný typ filtru pomocí charakteristické frekvence. Tímto získáme přenosovou funkci analogového prototypu *H(s)*, překreslí se „graf H(s)“ a opět se vypíší příslušné koeficienty a vztahy.

Poslední tlačítko v návrhu **[5]BTZ** převede navržený analogový prototyp *H(s)* na *H(z)* v z-rovině. „Graf H(z)“ přejde do finální podoby. Pokud na něj klikneme levým tlačítkem myši, otevře se nové okno **figure** a umožní nám grafickou observaci vykresleného tolerančního schématu a průběhu přenosové funkce H(z) na celé obrazovce jako na obr.6.8

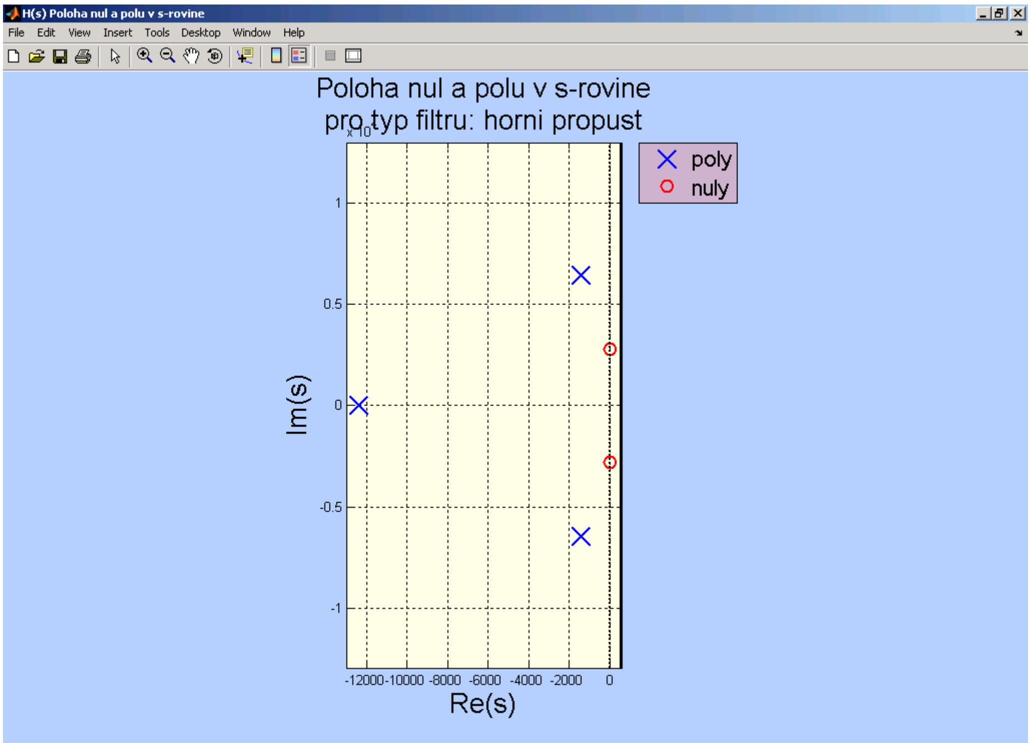
Trojice tlačítek označených: **ZP H(p)**, **ZP H(s)** a **ZP H(z)** umožňuje po výpočtu koeficientů v závěru konkrétního kroku zobrazit polohu nulových bodů a pólů.



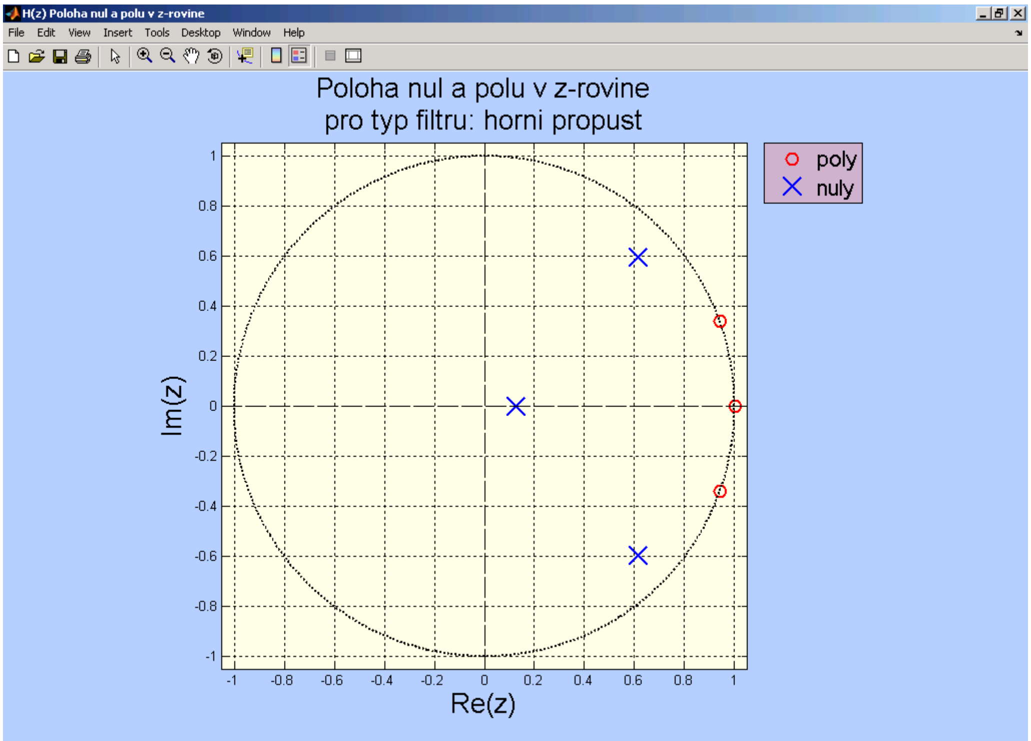
Obr. 6.8 Ukázka grafu přenosové funkce *H(z)*



Obr. 6.9 Výstup tlačítka „ZPHP“



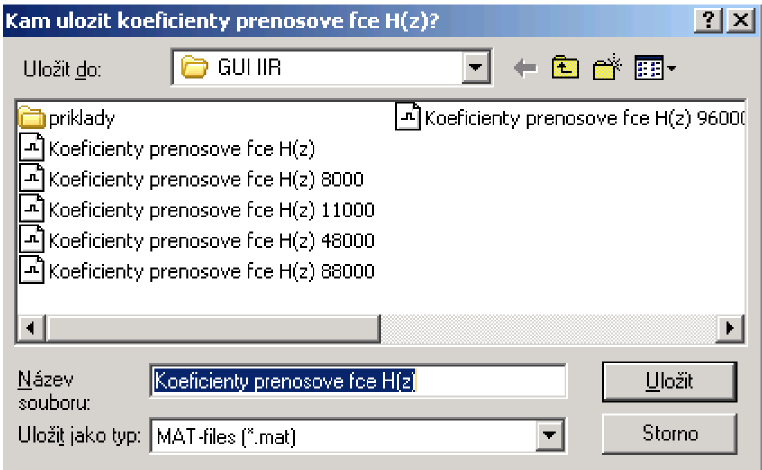
Obr. 6.10 Výstup tlačítka „ZPHS“



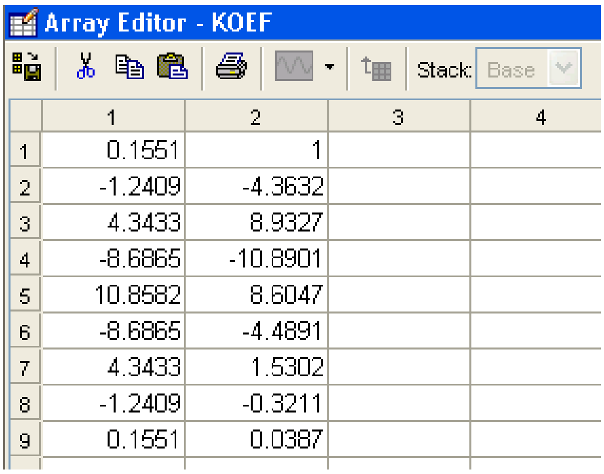
Obr. 6.11 Výstup tlačítka „ZPHZ“

Sestava tlačítek: **SPUST**, **ULOZ**, **SMAZ** a **KONEC** plní tyto úkoly v tomto pořadí:

1. postupně spouští jednotlivé kroky návrhu v definovaných časových úsecích;
2. uloží koeficienty přenosové funkce *H(z)*viz obr. X.y a ;
3. provede výmaz všech položek jako po prvním spuštění;
4. ukončí GUI.



Obr. 6.12 Vzhled okna pro uložení do souboru



Obr. 6.13 Ukázka načtení koeficientů přenosové funkce do **workspace**

6.1.3 Detailní popis funkčních celků programu:

Vytvoření a otevření GUI zabezpečuje hlavní funkce pojmenovaná „IIRF.m“. Ta zajistí vytvoření grafického objektu **figure** a všech dalších grafických objektů. Nyní se čeká na zásah uživatele a takto pomocí příkazu **switch** dojde k rozřazení do příslušných skriptů nebo funkcí na základě položky „callback“ obsažené v každém objektu. Zmíněný „callback**“** představuje v podstatě zpětnou vazbu, neboli jaký parametr obsahuje, takový **case** se bude vykonávat.

Použité skripty a funkce lze podle účelu rozčlenit do těchto skupin:

- skripty spjaté s funkčními tlačítky:
  - vykonávají se bezprostředně po stisku;
  - „RUNF.m“: „1,krok“
  - „PREWARPF.m“: „2.krok“
  - „HPF.m“: „3.krok“
  - „HSF.m“: „4.krok“
  - „BTZF.m“: „5.krok“
  - „ZPHP.m“: zobrazení nul a pólů H(p)
  - „ZPHS.m“: zobrazení nul a pólů H(s)
  - „ZPHZ.m“: zobrazení nul a pólů H(z)
  - „RUNALLF.m“: kompletní návrh krok po kroku
  - „SAVEF.m“: ukládání koeficientů
  - „CLEARF.m“: uvede program do výchozího stavu
- výkonné a podpůrné skripty
  - navazují na předchozí, tvoří pevné jádro návrhu;
  - „READF.m“: načtení vstupních parametrů
  - „PREWF.m“: předkreslení kmitočtových os
  - „DPNF.m“: podrobněji popsán níže
  - „BUTTERF.m“: návrh koeficientů
  - „CH1F.m“: návrh koeficientů

- „CH2F.m“: návrh koeficientů
- „ELLIPF.m“: návrh koeficientů
- „FILLHPF.m“: vykreslení tolerančního schématu H(p)
- „FILLHSF.m“: vykreslení tolerančního schématu H(s)
- „FILLHZF.m“: vykreslení tolerančního schématu H(z)
- „TEXTAXIF.m“: vyobrazení vzorců k 1.kroku
- „TEXTAXIIF.m“: vyobrazení vzorců k 2.kroku
- „TEXTAXIIIF.m“: vyobrazení vzorců k 3.kroku
- „TEXTAXIVF.m“: vyobrazení vzorců k 4.kroku
- „TEXTAXVF.m“: vyobrazení vzorců k 5.kroku
- „VZORCEIIF.m“: vyobrazení rozšířených vzorců
- „REINITAXIF.m“: restart vzorců k 1.kroku
- „REINITAXIIF.m“: restart vzorců k 2.kroku
- „REINITAXIIIF.m“: restart vzorců k 3.kroku
- „REINITAXIVF.m“: restart vzorců k 4.kroku
- „REINITAXVF.m“: restart vzorců k 5.kroku
- „REINITHPF.m“: restart H(p)
- „REINITHSF.m“: restart H(s)
- „REINITHZF.m“: restart H(z)
- „FIGGRAFF.m“: velký výstupní graf
- „PRINTF.m“: tisk **figure** do souboru
- „KEYPRESSF.m“: ovládání pomocí klávesnice
- „MYINIT.m“: přednastavení hodnot pro demonstraci
- „NOVYRAD.m“: aktualizace řádu v editačním poli

➢ návrhové aproximační funkce

- „butt.m“: Butterworth aproximace
- „cheb1.m“: Čebyševova aproximace
- „cheb2.m“: Čebyševova inverzní aproximace
- „cauer.m“: Cauerova eliptická aproximace. Viz kap 3.3

➢ matematické funkce viz kap. 3.3.4
    - „uelint.m“:
    - „nelint.m“;
    - „jacob.m“;

DPNF představuje klíčový výkonný orgán celého programu. V tomto skriptu je sdružen veškerý návrh filtru. Vstupují do něj parametry připravené ve 2. kroku. Na základě typu filtru využitím dopředné transformace se stanoví OMEGAS. Na základě typu aproximace využívá tyto skripty, pro které je nadřízeným prvkem:

- „BUTTERF.m“
- „CH1F.m“
- „CH2F.m“
- „ELLIPF.m“.

Získáme NDP. Dále se opět na základě typu filtru provede využitím MATLAB příkazů např. pro dolní propust „[tfbs,tfas]=lp2lp(tfbp,tfap,WN)“ tzv. zpětná kmitočtová transformace a posléze příkazem „[tfbz,tfaz]=bilinear(tfbs,tfas,FVZ)“ bilineární transformace.

Výsledkem činnosti skriptu DPNF jsou koeficienty pro všechny tři přenosové funkce, které se uloží do položky „userdata“.

Skripty „HSF.m“ a „BTZ.m“ plní roli pouze zobrazovací a vykreslovací a patřičné koeficienty si natahují s položek „userdata“ před tím naplněných činností DPNF.

Globální proměnné:

- „OMEGAS“: parametr NDP
- „POLOMER“: poloměr kružnice u Butterworthovy aproximace
- „ael“, „bel“: poloosy elipsy u Čebyševovy aproximace

GUI lze ovládat i pomoci klávesnice. Pro jednotlivé funkční tlačítka existují protějšky na klávesnici.

Klávesové zkratky pro ovládání GUI:

- klávesa „d“ : přednastavení hodnot pro demonstraci;
- klávesa „a“ : kompletní návrh krok po kroku;
- klávesa „u“: ukládání koeficientů;
- klávesa „v“ : velký výstupní graf;
- klávesa „s“: uvede program do výchozího stavu;
- klávesa „k“ : pro ukončení aplikace;
- klávesa „1“ : „1,krok“;
- klávesa „2“ : „2,krok“;
- klávesa „3“ : „3,krok“;
- klávesa „4“ : „4,krok“;
- klávesa „5“ : „5,krok“;
- klávesa „`“ : vyobrazení rozšířených vzorců.
- klávesa „t“ : pro tisk **figure** do souboru.

6.1.4 Dílčí závěr:

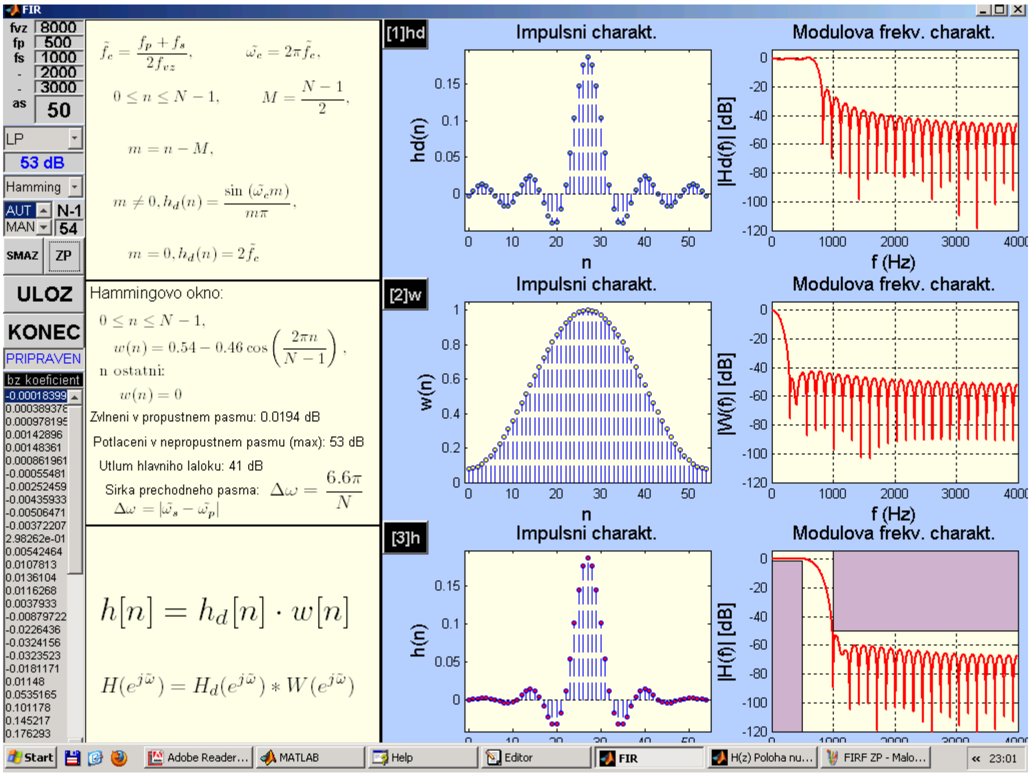
Uvedené zpracované GUI umožňuje provádět návrh číslicových filtrů IIR v aproximacích a typech filtrů zmíněných v kap. 3.2.1 Mezi limitující faktor jednoznačně patří přítomnost zkreslení a nestability výsledného filtru, která znehodnotí návrh, pokud navrhneme filtr s vysokým řádem (vyšší než 9). Nejvíce je takto postižena Butterworthova aproximace. Tato nedokonalost je způsobena především typem realizace výsledné přenosové funkce ve formě podílu čitatele a jmenovatele. Uvedený nedostatek lze odstranit využitím jiné formy vyjádření přenosové funkce, nejlépe ve formě sériové (kaskádní), která je podstatně odolnější z hlediska dopadu tzv. konečné délky slova viz kap. 5.3.3.

6.2 Tvorba GUI FIR

Po krátké rozvaze jsem přistoupil ke zpracování GUI pro metodiku návrhu filtrů FIR.

Uvedené GUI je opět vytvořeno osvědčenou metodou „Switched Board Programming“.

Kompletní pohled na aplikaci s provedením návrhu filtru:



Obr. 6.14 Celkový pohled na GUI FIR

6.2.1 Rozvržení ovládacích prvků

Program GUI využívá:

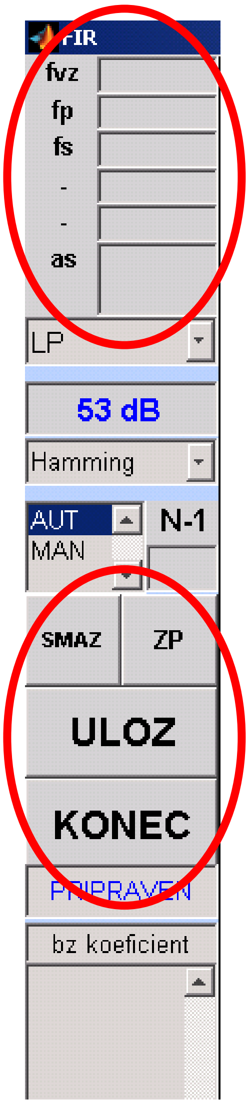
- hlavní a 2 dočasná okna **figure**;
- **sedm editačních polí** pro specifikaci tolerančního schématu a zobrazení indikace činnosti GUI, minimálního doporučeného řádu filtru nebo pro jeho manuální zadání;

- **7 tlačítek** pro uložení koeficientů přenosové funkce *H(z)* do souboru na disk, tlačítko pro kompletní vymazání vstupních parametrů, všech grafů, vzorců a koeficientů, pro zobrazení polohy nul a pólů přenosové funkce *H(z)*, tlačítka pro jednotlivé kroky návrhu a pro kompletní ukončení práce v GUI);
- **trojici výběrových menu** (umožňují volbu typu, volbu okna, pro volbu mezi automatickým nebo manuálním řádem filtru;
- **devět polí axes** (pro zobrazení grafů tolerančního schématu, impulsní charakteristiky, modulové frekvenční charakteristiky a použitých výpočetních vztahů);
- **jedno rolovatelné pole „listbox“** (pro zobrazení koeficientu přenosové funkce *H(z)*).

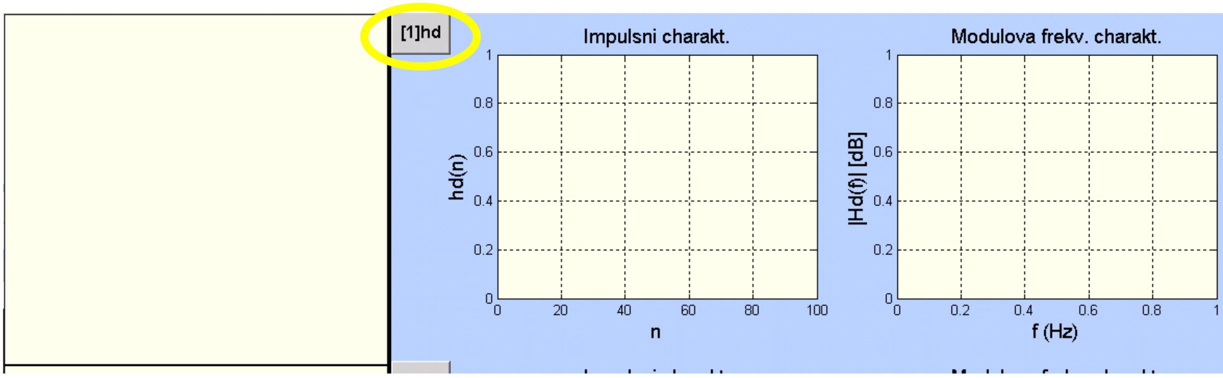
Hlavní okno na obr.6.14 lze z hlediska návrhu rozdělit na celkem 3 samostatné celky pokrývající jeho jednotlivé fáze:

1.celek tvoří objekty zajišťující opět základní běh, zkrácenou impulsní charakteristiku a obsahuje tyto následující komponenty, které spolu tvoří funkční celek nebo připravují hodnoty pro následující celek:

- soubor tlačítek pro běh programu označených modrou barvou
- indikátor činnosti/nečinnosti ve zeleném rámečku
- informační box vyjadřující útlum v oranžové barvě
- editační políčka a výběrové menu v sekci vstupních parametrů vyznačených červenou barvou
- tlačítko **[1]hd** barvou žlutou
- 3x pole axes
- 1x rolovatelné pole vyčleněné pro koeficienty



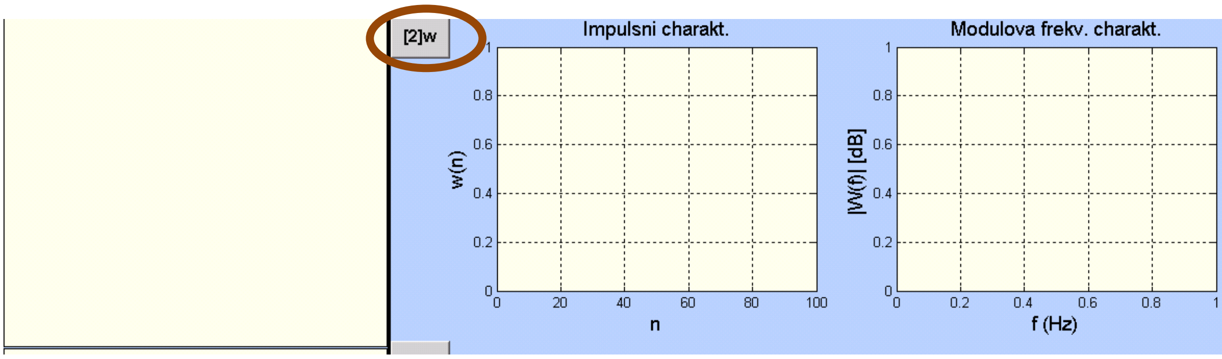
Obr. 6.15 Vstupní parametry GUI



Obr. 6.16 Vzhled 1. celku

2. celek tvoří objekty zajišťující popis okenní funkce a obsahuje tyto následující komponenty, které spolu tvoří funkční celek nebo opět připravují hodnoty pro následující celek:

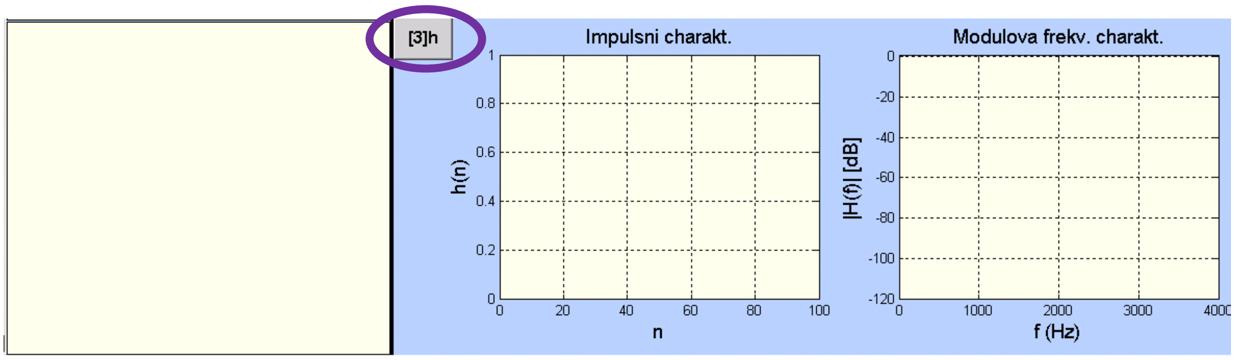
- tlačítko **[1]w** barvou hnědou
- 3x pole axes



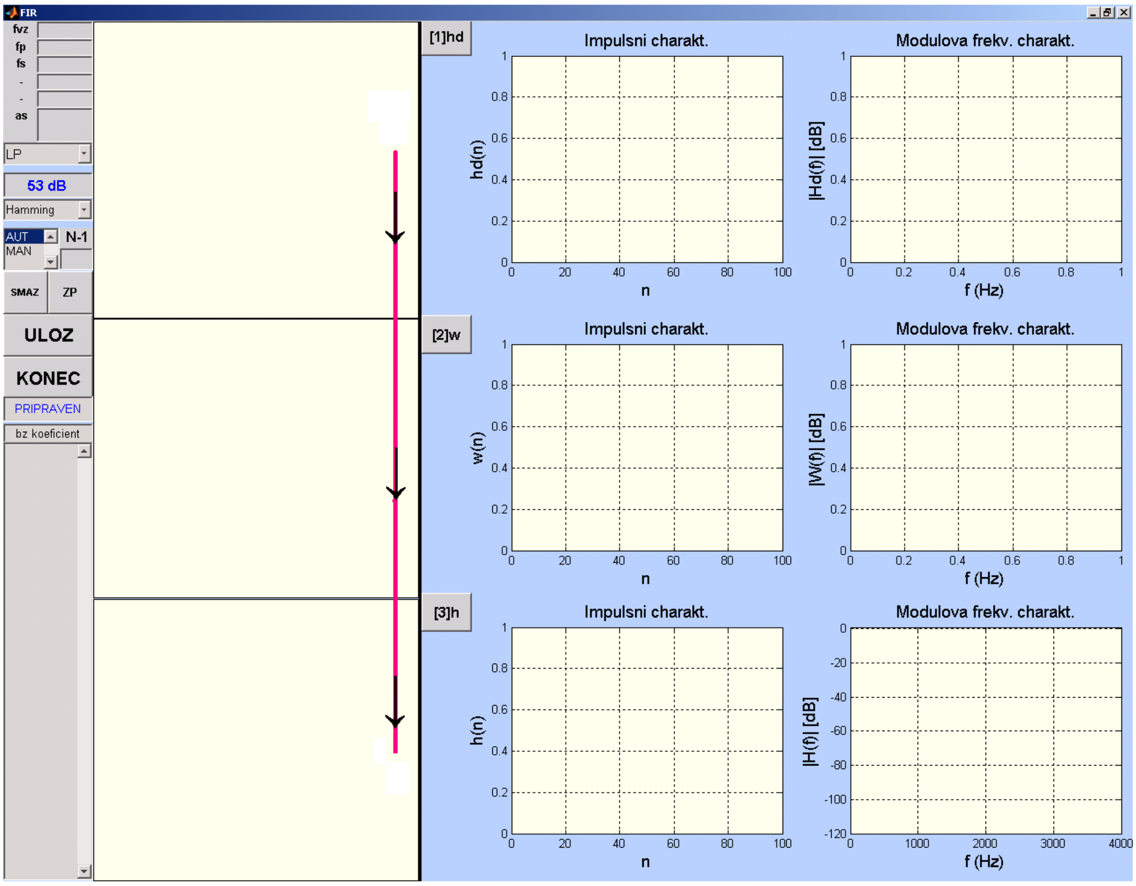
Obr. 6.17 Vzhled 2. celku

3. celek tvoří objekty zajišťující výslednou přenosovou funkci a obsahuje tyto následující komponenty:

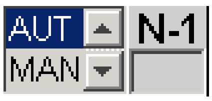
- tlačítko **[3]h** označené fialovou barvou
- 3x pole axes



Obr. 6.18 Vzhled 3. celku



Obr. 6.19 Vzhled GUI včetně návrhové šipky



Obr. 6.20 Přepínač řádu filtru

6.2.2 Popis funkce GUI

Grafické uživatelské rozhraní můžeme spustit přímo z příkazového řádku zadáním „FIRF“ nebo dvojklikem myší do aktuální složky v prostředí MATLAB na soubor FIRF.m. Otevře se hlavní okno **figure** a v levém horním rohu můžeme postupně v jednotkách hertz zadat:

- vzorkovací frekvenci f_{vz},
- frekvence pro propustné a nepropustné pásmo, nastavujeme opět od shora dolů, v souladu se značením a v závislosti na typu filtru (dolní, horní, pásmová propust nebo pásmová zádrž), který většinou navolíme úplně úplně na začátku.
- dále v decibelech stanovíme požadovaný útlum pro nepropustné pásmo, a dostaneme tip jakou okenní funkci na základě velikosti útlumu zvolit. Bude zobrazena v prostředním okně pro vzorce okenních funkcí.

Návrh filtru zajišťují tři funkční tlačítka, která jsou úzce spjata s jednotlivými kroky, které se provádějí ve směru návrhové šipky od viz obr. 6.19.

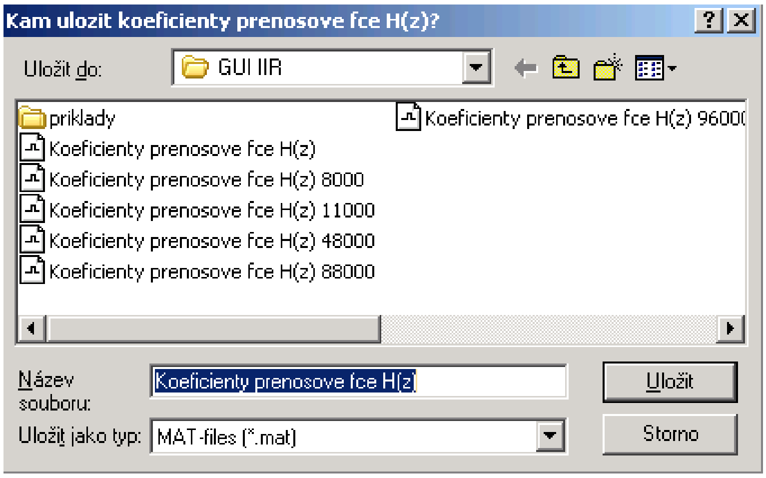
Po úvodním nadefinování tolerančního schématu můžeme začít s návrhem filtru, který odstartujeme tlačítkem [**1]hd**. Vykoná se 1. krok návrhu, který zajistí výpočet zkrácené impulsní charakteristiky dle odhadnuté délky filtru *N*, (*N-1*) je řád filtru. Pokud nebude délka filtru dostatečná, zkorigujeme ji ručně. Výsledná impulsní charakteristika se vykreslí do patřičného grafu. Graf napravo od ní nám zobrazí modulovou frekvenční charakteristiku. Současně se vyobrazí i vzorec pro stanovení ideální impulsní charakteristiky frekvence [rad], jenž využívá následující krok. Výsledný vektor koeficientů uloží do globální proměnné **hd**.

[2]w tlačítko provede výpočet impulsní charakteristiky zvolené okenní funkce. Výsledky vyobrazí analogicky do patřičných grafů jako v případě 1. kroku návrhu. Výsledný vektor koeficientů uloží do globální proměnné **w**.

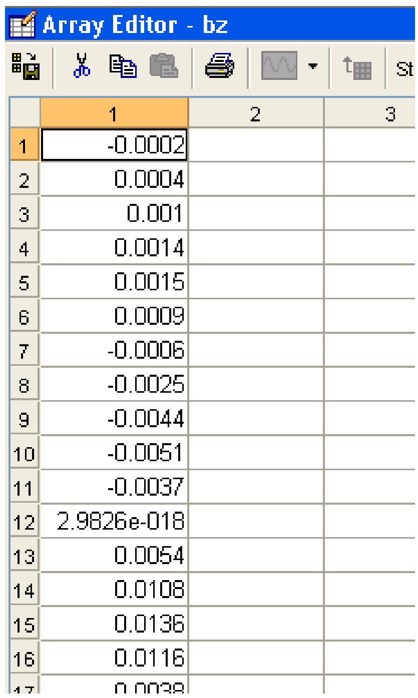
Stisknutí tlačítka **[3]h** provede vynásobení zkrácené impulsní charakteristiky zvoleným oknem a získáme výsledný vektor **h** koeficientů přenosové funkce *H(z)*, který se uloží do userdata.

Sestava tlačítek: **ULOZ**, **SMAZ** a **KONEC** plní tyto úkoly v tomto pořadí:

1. uloží koeficienty přenosové funkce *H(z)*viz obr. 6.21;
2. provede výmaz všech položek jako po prvním spuštění;
3. ukončí GUI.



Obr. 6.21 Vzhled okna pro uložení do souboru



Obr. 6.22 Ukázka načtení koeficientů do **workspace**

6.2.3 Detailní popis funkčních celků programu:

Vytvoření a otevření GUI zabezpečuje hlavní funkce pojmenovaná „FIRF.m“. Ta zajistí vytvoření grafického objektu **figure** a všech dalších grafických objektů.

Použité skripty a funkce lze podle účelu rozčlenit do těchto skupin:

- skripty spjaté s funkčními tlačítky:
    - vykonávají se bezprostředně po stisku;
    - „RUNF.m“: „1,krok“
    - „WND.m“: „2.krok“
    - „HF.m“: „3.krok“
    - „ZPHZ.m“: zobrazení nul a pólů H(z)
    - „SAVEF.m“: ukládání koeficientů
    - „CLEARF.m“: uvede program do výchozího stavu
- výkonné a podpůrné skripty
    - navazují na předchozí, tvoří pevné jádro návrhu;
    - „READF.m“: načtení vstupních parametrů
    - „BARLETTF.m“: okenní funkce Barlett
    - „BLACKF.m“: okenní funkce Blackman
    - „HAMMF.m“: okenní funkce Hamming
    - „HANNF.m“: okenní funkce von Hann
    - „KAISF.m“: okenní funkce Kaiser
    - „RECTF.m“: okenní funkce obdelníková
    - „VOLBAWNDF.m“: návrh okna
    - „FILLHZF.m“: vykreslení tolerančního schématu H(z)
    - „TEXTVZIF.m“: vyobrazení vzorců k 1.kroku
    - „TEXTVZIIF.m“: vyobrazení vzorců k 2.kroku
    - „TEXTVZIIIF.m“: vyobrazení vzorců k 3.kroku
    - „REINITVZIF.m“: restart vzorců k 1.kroku
    - „REINITVZIIF.m“: restart vzorců k 2.kroku

- „REINITVZIIIF.m“: restart vzorců k 3.kroku
- „REINITWNDF.m“: restart grafu imp. char. 2.kroku
- „REINITWNDZF.m“: restart grafu frekv.char. k 2.kroku
- „REINITHDF.m“: restart grafu imp. char. 1.kroku
- „REINITHDZF.m“: restart grafu frekv.char. 1.kroku
- „REINITHF.m“: restart grafu imp. char. 3.kroku
- „REINITHZF.m“: restart grafu frekv.char. 3. kroku
- „FIGGRAFF.m“: velký výstupní graf
- „PRINTF.m“: tisk **figure** do souboru
- „KEYPRESSF.m“: ovládání pomocí klávesnice
- „MYINIT.m“: přednastavení hodnot pro demonstraci

➢ Výpočet impulsní charakteristiky
- „ideal_LP.m“: dolní propust
- „ideal_HP.m“: horní propust
- „ideal_BP.m“: pásmová propust
- „ideal_BS.m“: pásmová zádrž

Skripty „HSF.m“ a „BTZ.m“ plní roli pouze zobrazovací a vykreslovací a patřičné koeficienty si natahují s položek „userdata“ před tím naplněných činností DPNF.

Globální proměnné:

- „w“: vektor koeficientů okna
- „hd“: vektor koeficientů zkrácené impulsní charakteristiky

GUI lze ovládat i pomoci klávesnice. Pro jednotlivé funkční tlačítka existují protějšky na klávesnici.

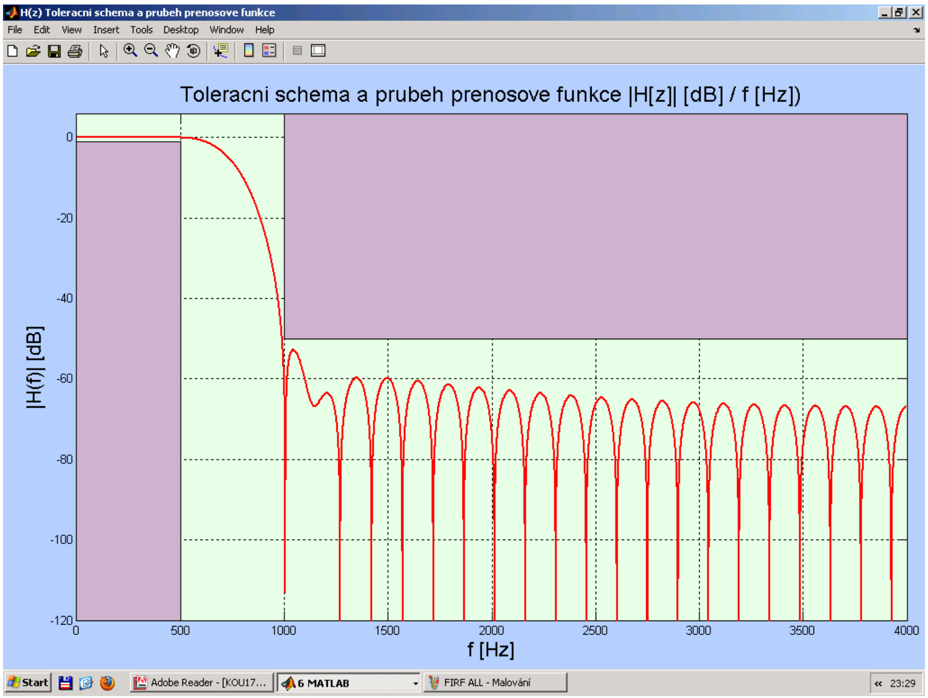
Klávesové zkratky pro ovládání GUI:

- klávesa „d“ : přednastavení hodnot pro demonstraci;
- klávesa „u“: ukládání koeficientů;
- klávesa „v“ : velký výstupní graf;
- klávesa „s“: uvede program do výchozího stavu;
- klávesa „k“ : pro ukončení aplikace;
- klávesa „1“ : „1,krok“;
- klávesa „2“ : „2,krok“;
- klávesa „3“ : „3,krok“;
- klávesa „t“ : pro tisk **figure** do souboru.

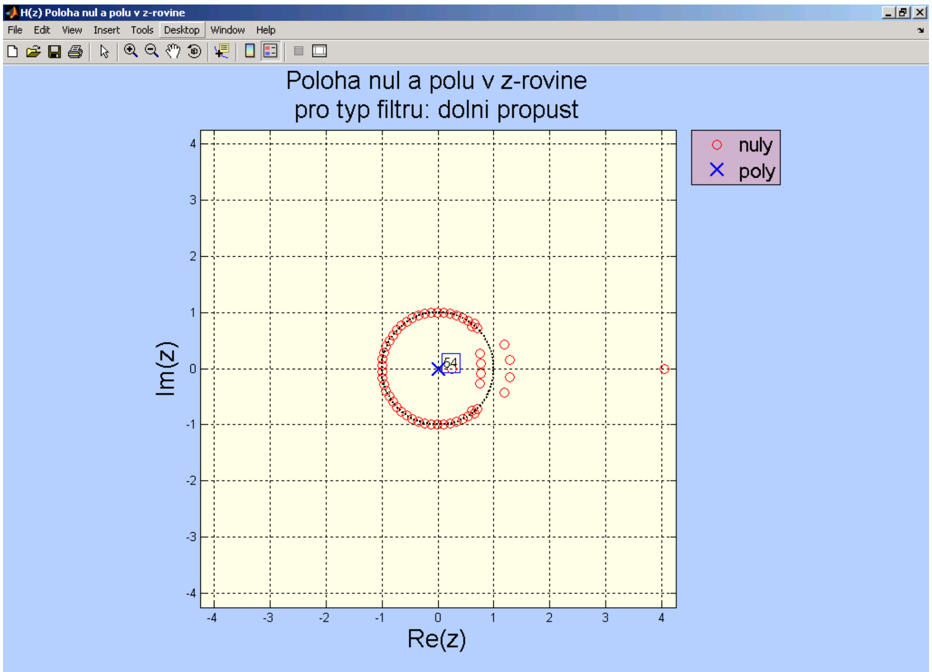
6.2.4 Dílčí závěr:

Uvedené zpracované GUI umožňuje provádět návrh číslicových filtrů FIR využívající okenních funkcí zmíněných v kap. 4.2.1.

Mezi limitující faktor tohoto GUI patří nemožnost zobrazení nulových bodů a pólů, pokud je délka filtru v řádu tisíců. Paměť systému je v tomto případě nedostatečná.



Obr. 6.23 Ukázka grafu přenosové funkce *H(z)*



Obr. 6.24 Výstup tlačítka „ZP“

V kapitole 6 bylo čerpáno z:[2],[4],[6]

Závěr

Cílem této práce bylo vytvoření grafického uživatelského rozhraní v prostředí MATLAB pro návrh filtrů IIR a FIR s jednotlivými kroky návrhu. Vytvořil jsem dvě velmi podobné interaktivní aplikace. První se týká IIR, druhá nás seznámí s problematikou filtrů FIR.

GUI IIR pokrývá jednotlivé kroky návrhu číslicového filtru IIR metodou využívající analogový prototyp. Uživatel vyplní vstupní parametry a může jednotlivé kroky libovolně spouštět a sledovat tak jednotlivé fáze návrhu. Lze zobrazit rozložení nul a pólů, výpočtové vzorce, koeficienty přenosových funkcí a rovněž křivky jednotlivých modulových kmitočtových charakteristik. Koeficienty přenosové funkce H(*z)* lze uložit na disk a provádět s nimi další zpracování . Dále umí GUI vytisknout do souboru výstupní graf přenosové funkce H(z), který je zobrazen do samostatného okna figure.

GUI FIR demonstruje návrh filtrů typu FIR využitím okenních funkcí. Většina ovládacích prvků je okopírovaná z GUI IIR, takže i ovládání bude velmi podobné.

Přínosem vytvořených GUI může být jejich praktické využití ve výuce, kde může usnadnit a zlepšit pochopení souvislostí a kde se může i úplný začátečník seznámit s návrhem krok po kroku.

Použitá literatura

[1] SMÉKAL, Z, *Číslicové filtry,* Vydání 1, Brno: Nakladatelství VUT, 1993,136 s, ISBN 80-214-0500-7,

[2] JAN, J, *Číslicová filtrace, analýza a restaurace signálů,* Vydání 2, Brno: Nakladatelství VUTIUM, 2002, 427 s, ISBN 80-214-1558-4,

[3] DAVÍDEK, V,; LAIPERT, M,; VLČEK, M, *Analogové a číslicové filtry,* Vydání 2, Praha: Nakladatelství ČVUT, 2006, 345 s, ISBN 80-01-03026-1,

[4] ZAPLATÍLEK, K,; DOŇAR, B, *MATLAB: tvorba uživatelských aplikací,* Vydání 1, Praha: BEN - technická literatura, 2004, 215 s, ISBN 80-7300-133-0,

[5] OPPENHEIM, Alan V,; SCHAFER, Ronald W,; BUCK, John R, *Discrete-Time Signal Processing,* Vydání 1, New Jersey: Prentice-Hall, 1999, 870 s, ISBN 0-13-198842-5,

[6] Portál VŠB: Zpracování číslicových signálů-cvičení [online], 2012 [cit, 2012-05-08], PCM2902, Dostupné z WWW:

<http://homel,vsb,cz/~ska109/ZCS/000_Literatura/ZCS_skripta,pdf>

Seznam obrázků

Seznam tabulek

Přílohy

Zdrojové kódy aplikací jsou uloženy ve dvou složkách na přiloženém nosiči CD ROM. První aplikace IIR je uložená v souboru s názvem IIRF.m a spolu se všemi příslušnými skripty a funkcemi se nachází ve složce GUI IIR. Druhá aplikace FIR je uložená v souboru s názvem FIRF.m a spolu se všemi příslušnými skripty a funkcemi ji lze nalézt ve složce GUI FIR.